```
    515 000002                        .BLKW   1               ; C.STS
    516 000004              D$TADR::.BLKW   5.              ; RAD50 STRING OF 1.-15. DECIMAL DIGITS
    517 000016              D$TLIN::.BLKW   1               ; LINE ID:: RAD50 PROCESS NAME
    518 000020                      .BLKW   3               ;          CONTROLLER, UNIT, TRIB NUMBERS
    519 000026              D$TSTA::.BLKW   1               ; =0 --> OFF, >0 --> ON
    520 000030              D$THSH::.BLKW   1               ; HASH TABLE SIZE (32.-512. ENTRIES)
    521 000032              D$TCT:: .BLKW   1               ; COUNTER TIMER VALUE (0.-65535.)
    522 000034              D$TNET::.BLKW   2               ; RAD50 NETWORK NAME
    523         000040      D$TLEN==.
    524 002054                      .PSECT
    525
    526                     ;****
    527                     ;
    528                     ; DEFINE THE OUTGOING CHANNEL DEFINITION BLOCK
    529                     ;
    530 000040                      .ASECT
    531         000000              .=0
    532 000000                      .BLKW   1               ; C.LNK
    533 000002                      .BLKW   1               ; C.STS
    534 000004              C$HLO::.BLKW    1               ; LOW END OF RANGE
    535 000006              C$HHI::.BLKW    1               ; HIGH END OF RANGE
    536         000010      C$HLEN==.
    537 002054                      .PSECT
    538
    539                     ;****
    540                     ;
    541                     ; DEFINE THE CUG PARAMETER BLOCK
    542                     ;
    543 000010                      .ASECT
    544         000000              .=0
    545 000000                      .BLKW   1               ; C.LNK
    546 000002                      .BLKW   1               ; C.STS
    547 000004              C$UNAM::.BLKW   2               ; CUG NAME
    548 000010              C$UCUG::.BLKW   1               ; GROUP NUMBER
    549 000012              C$UFLG::.BLKB   1               ; FLAGS BYTE
    550         000013      C$ULEN==.
    551
    552                     ;*****
    553                     ;
    554                     ; DEFINE THE SVC PARAMETER BLOCK
    555                     ;
    556 000013                      .ASECT
    557         000000              .=0
    558 000000                      .BLKW   1               ; C.LNK
    559 000002                      .BLKW   1               ; C.STS
    560 000004              S$VNAM::.BLKW   2               ; NAME (XX.YY OF DLM-XX.YY)
    561 000010              S$VPSZ::.BLKW   1               ; PACKET SIZE
    562 000012              S$VWSZ::.BLKW   1               ; WINDOW SIZE
    563 000014              S$VFLG::.BLKW   1               ; FLAGS
    564 000016              S$VRCL::.BLKW   1               ; MAXIMUM RECALLS
    565 000020              S$VTIM::.BLKW   1               ; RECALL TIMER
    566 000022              S$VDTE::.BLKW   5               ; RAD50 OF DTE ADDRESS
    567 000034              S$VOWN::.BLKW   1               ; RAD50 OWNER OF SVC
    568         000036      S$VLEN==.
    569 002054                      .PSECT
    570
    571                     ;*****
```

```
   1150       040000                          CS.UNI  ==  40000      ; =1 IF UNIT SPECIFIED IN LINE ID
   1151       020000                          CS.TRB  ==  20000      ; =1 IF TRIB SPECIFIED IN LINE ID
   1152       010000                          CS.FLG  ==  10000      ; =1 IF FLAGS SPECIFIED
   1153                                 ;
   1154                                 ; TEMPLATE SPECIFIC STATUS FLAGS
   1155                                 ;
   1156       000400                          CS.TOP  ==    400      ; =1 IF TOP DOWN LOADING (PAR$DF)
   1157
   1158       000400                          CS.THI  ==    400      ; =1 IF TRANSPORT HI SPECIFIED (PSN$DF)
   1159       001000                          CS.TLO  ==   1000      ; =1 IF TRANSPORT LO SPECIFIED (PSN$DF)
   1160
   1161       000400                          CS.NET  ==    400      ; =1 IF NETWORK NAME SPECIFIED (DTE$DF)
   1162
   1163       000400                          CH$HI   ==    400      ; =1 IF HIGH CHANNEL SPECIFIED (CHN$DF)
   1164
   1165       000400                          PV$OWN  ==    400      ; =1 IF PVC OWNER PROCESS SPECIFIED (PVC$DF)
   1166       001000                          PV$PSZ  ==   1000      ; =1 IF PVC MAX BLOCK SIZE SPECIFIED (PVC$DF)
   1167       002000                          PV$WSZ  ==   2000      ; =1 IF PVC WINDOW SIZE SPECIFIED (PVC$DF)
   1168
   1169       000001                          CU$BIL  ==      1      ; FLAG BIT IN C$UFLG FOR BILATERAL GROUP
   1170
   1171       000400                          CS.TSK  ==    400      ; =1 IF DESTINATION TASKNAME SPECIFIED (DST$DF)
   1172
   1173       000400                          CS.NUM  ==    400      ; =1 IF DESTINATION DTE ADDR SPECIFIED (DSA$DF)
   1174       001000                          CS.SHI  ==   1000      ; =1 IF HIGH SUBADDRESS RANGE SPECIFIED (DSA$DF)
   1175       002000                          CS.SLO  ==   2000      ; =1 IF LOW SUBADDRESS RANGE SPECIFIED (DSA$DF)
   1176       004000                          DA$CUG  ==   4000      ; =1 IF CLOSED USER GROUP SPECIFIED (DSA$DF)
   1177
   1178       000400                          CS.MSK  ==    400      ; =1 IF DESTINATION CALL MASK SPECIFIED (DSC$DF)
   1179       001000                          CS.VAL  ==   1000      ; =1 IF DESTINATION CALL VALUE SPECIFIED (DSC$DF)
   1180
   1181       000400                          ND$EXA  ==    400      ; =1 IF AREA SPECIFIED IN EXEC ADDRESS (NOD$DF)
   1182       001000                          ND$HOA  ==   1000      ; =1 IF AREA SPECIFIED IN HOST ADDRESS (NOD$DF)
   1183
   1184       000400                          RE$ARE  ==    400      ; =1 IF AREA SPECIFIED (REM$DF)
   1185
   1186       000400                          SL$CTM  ==    400      ; =1 IF COUNTER TIMER SPECIFIED (SLT$DF)
   1187       001000                          SL$TYP  ==   1000      ; =1 IF TYPE SPECIFIED (SLT$DF)
   1188       002000                          SL$HTM  ==   2000      ; =1 IF HELLO TIMER SPECIFIED (SLT$DF)
   1189       004000                          SL$RPR  ==   4000      ; =1 IF ROUTER PRIORITY SPECIFIED (SLT$DF)
   1190       010000                          SL$BRA  ==  10000      ; =1 IF BROADCAST ROUTERS SPECIFIED (SLT$DF)
   1191
   1192       000400                          SV$PSZ  ==    400      ; =1 IF SVC PACKET SIZE SPECIFIED (SVC$DF)
   1193       001000                          SV$WSZ  ==   1000      ; =1 IF SVC WINDOW SIZE SPECIFIED (SVC$DF)
   1194       002000                          SV$DTE  ==   2000      ; =1 IF DTE NUMBER SPECIFIED (SVC$DF)
   1195
   1196       000400                          UN$SCS  ==    400      ; SECONDARY CSR SPECIFIED (UNT$DF)
   1197       001000                          UN$CST  ==   1000      ; CIRCUIT COST SPECIFIED (UNT$DF)
   1198       002000                          UN$DPR  ==   2000      ; DEAD POLLING RATION SPECIFIED (UNT$DF)
   1199       004000                          UN$PCH  ==   4000      ; EMULATOR CHARACTERISTICS SPECIFIED (UNT$DF)
   1200
   1201       000400                          ST$HTM  ==    400      ; HELLO TIMER SPECIFIED (STA$DF)
   1202
```

```
                                                                   B  3
CFEIMP      CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 8
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE              REFERENCES

P$VOWN      000020  G      #9-502
P$VPSZ      000014  G      #9-500
P$VSTA      000023  G      #9-504
P$VWSZ      000016  G      #9-501
Q.IOAE   =  000012  G       6-95
Q.IOFN   =  000002  G       6-95          6-95          7-127         7-128
Q.IOPL   =  000014  G       6-95
Q.IOSB   =  000010  G       6-95          6-95
RE$ARE   =  000400  G      #12-1184
RN.DCT   =  000100  G      #11-902        11-914
RN.DIA   =  020000  G      #11-909        11-915
RN.DMP   =  040000  G      #11-910        11-915
RN.DPA   =  000040  G      #11-901        11-914
RN.HAR   =  000400  G      #11-904        11-914
RN.HOS   =  000004  G      #11-898        11-914
RN.LOD   =  010000  G      #11-908        11-915
RN.NAM   =  000002  G      #11-897
RN.SCI   =  000200  G      #11-903        11-914
RN.SDV   =  000010  G      #11-899        11-914
RN.SLO   =  002000  G      #11-906        11-915
RN.SNV   =  000020  G      #11-900        11-914
RN.SPA   =  001000  G      #11-905        11-914
RN.TLO   =  004000  G         1-907       11-915
RN.XXX   =  077774  G      #11-915
R$DADR      000010  G      #9-661
R$DLEN   =  000022  G      #9-662
R$DNAM      000004  G      #9-660
R.VAR    =  ******  GX      6-105         6-111
SF.ENA   =  000100  G      #12-1083
SF.MFL   =  000040  G      #12-1084
SF.SER   =  000010  G      #12-1086
SF.SVC   =  000020  G      #12-1085
SL$BRA   =  010000  G      #12-1190
SL$CTM   =  000400  G      #12-1186
SL$HTM   =  002000  G      #12-1188
SL$RPR   =  004000  G      #12-1189
SL$TYP   =  001000  G      #12-1187
SP.ACC   =  000001  G      #11-1021
SP.CIR   =  000004  G      #11-1023
SP.DAT   =  020000  G      #11-1034
SP.KNA   =  000002  G      #11-1022
SP.KPU   =  001000  G      #11-1030
SP.MLU   =  002000  G      #11-1031
SP.MOD   =  000040  G      #11-1026
SP.PLU   =  000020  G      #11-1025
SP.PSW   =  000200  G      #11-1028
SP.SID   =  010000  G      #11-1033
SP.SLU   =  000010  G      #11-1024
SP.SPU   =  000400  G      #11-1029
SP.UID   =  000100  G      #11-1027
SP.XID   =  004000  G      #11-1032
ST$HTM   =  000400  G      #12-1201
```

```
                                                                   C  3
CFEIMP      CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 9
```

```
                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                ;****
                                ; LIBRARY MACRO CALLS
                                ;****

                                        .MCALL  GCL$,CLOSE$,DELET$,FINIT$

                                ;****
                                ; LOCAL DATA
                                ;****
                                        .ENABL  LC
000000  015     012     105     PRMPT:  .ASCIZ  <15><12>/Enter filename: /
000003  156     164     145
000006  162     040     146
000011  151     154     145
000014  156     141     155
000017  145     072     040
000022  000
                                        .EVEN
```

B 5

```
**FILE**ID**CFEOPN

  CCCCCCCC  FFFFFFFFFF  EEEEEEEEEE    000000    PPPPPPPP    NN      NN
  CCCCCCCC  FFFFFFFFFF  EEEEEEEEEE    000000    PPPPPPPP    NN      NN
CC          FF          EE          00      00  PP      PP  NN      NN
CC          FF          EE          00      00  PP      PP  NN      NN
CC          FF          EE          00      00  PP      PP  NNNN    NN
CC          FF          EE          00      00  PP      PP  NNNN    NN
CC          FFFFFFFF    EEEEEEEE    00      00  PPPPPPPP    NN  NN  NN
CC          FFFFFFFF    EEEEEEEE    00      00  PPPPPPPP    NN  NN  NN
CC          FF          EE          00      00  PP          NN    NNNN
CC          FF          EE          00      00  PP          NN    NNNN
CC          FF          EE          00      00  PP          NN      NN      ....
CC          FF          EE          00      00  PP          NN      NN      ....
  CCCCCCCC  FF          EEEEEEEEEE    000000    PP          NN      NN      ....
  CCCCCCCC  FF          EEEEEEEEEE    000000    PP          NN      NN      ....


LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

C 5

```
                              .SBTTL  MACRO DEFINITIONS

                      ;***
                      ; GLOBAL MACROS
                      ;***

                      ;****
                      ; LOCAL MACROS
                      ;****

                              .MACRO  PRINT   TEXT
                              MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                              CALL    $PRINT          ; PRINT MESSAGE
                              .ENDM   PRINT

                              .MACRO  ERROR$  TEXT
                      ;
                      ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                      ; MESSAGE STRING.
                      ;
                              .IF DIF <TEXT><R0>
                              MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                              .ENDC
                              CALL    $CFERR                  ; PRINT ERROR MESSAGE
                              .ENDM   ERROR$

                              .MCALL  FLTDF$,SAVRG,RESRG

 000000                       FLTDF$                          ; DEFINE FILTER BLOCK SYMBOLS
```

```
                                                                                B  7
CFEPRG      CREATED BY  MACRO  ON 29-JUN-85 AT 04:57       PAGE 2
SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL   VALUE               REFERENCES

PURLOG     000360 R           5-108       #9-231
PUROBJ     000226 R           5-106       #8-178
P$VCT    = ******   GX      *11-386
P$VLEN   = ******   GX       11-349
P$VNAM   = ******   GX       11-350
P$VOWN   = ******   GX       11-368       11-384
QREM       001244 R           8-196        9-289      #12-417
SNKTBL     000202 R          #5-120        9-255
S$LCTT   = ******   GX      *10-314
TEMP       000212 R          #5-126       *9-234       9-235      *9-242
WC.EVE   = ******   GX        9-233        9-271
WC.LOG   = ******   GX        9-244        9-295
WC.OBJ   = ******   GX        8-189        8-206
X$LENX   = ******   GX       11-376       11-377
$BYTMN   = ******   GX       *8-200       *9-293      *11-376
$BYTXT   = ******   GX       *8-201       *9-294      *11-377
$CFERR   = ******   GX        8-212       11-397       11-399
$CFPRG     000214 RG         #7-161
$CIRID   = ******   GX       11-356       11-358
$CLASS   = ******   GX       *9-236        9-265
$COMP    = ******   GX        7-162       11-335       11-353
$EVENT   = ******   GX       *9-238       *9-239       *9-240      *9-241      *9-274      *9-275      *9-276      *9-277       9-278
$FLAGS   = ******   GX       11-338       11-340       11-366
$KNEVT   = ******   GX        9-234
$OPRND   = ******   GX        9-231        9-247        9-250       9-297      11-361
$PUREM   = ******   GX        5-102
$PUSNA   = ******   GX        5-115
$PUXAC   = ******   GX        5-113
$PUXPR   = ******   GX        5-111
$PUXSV   = ******   GX        5-112
$PUXS9   = ******   GX        5-114
$RLBF    = ******   GX        8-199        9-292       11-374
$TMLST   = ******   GX        8-179        9-259       10-311      11-345     *11-372     *12-420
$TNUM    = ******   GX        8-191
$WILD    = ******   GX        8-189        8-206       *9-233       9-244       9-271       9-295
$XXAVL   = ******   GX        8-198        9-291       11-373
.FNSLT   = ******   GX       10-312
```

```
                                                                                C  7
CFEPRG      CREATED BY  MACRO  ON 29-JUN-85 AT 04:57       PAGE
```

```
                                  .SBTTL  PURGE COMPONENTS
                          ;
                          ; PURGE COMPONENTS
                          ;
000000                            STATE$  PURGE,  &P$PUR
000000                             TRAN$  !KNOBJ, PUROPT   ,,CO.KNO,$COMP
000000                             TRAN$  !OBJID, PUROPT   ,,CO.OBJ,$COMP
000000                             TRAN$  !KNNOD, PURNOD   ,,CO.KNN,$COMP
000000                             TRAN$  !NODID, PURNOD   ,,CO.NOD,$COMP
000000                             TRAN$  !LOGID, LOGOPT   ,,CO.LOG,$COMP
000000                             TRAN$  !KNLOG, LOGOPT   ,,CO.KLG,$COMP
000000                             TRAN$  !LINID, PURLIN   ,,CO.LIN,$COMP
000000                             TRAN$  !KNLIN, PURLIN   ,,CO.KNL,$COMP
000000                             TRAN$  !KNCIR, PURCIR   ,,CO.KNC,$COMP
000000                             TRAN$  !CIRID, PURCIR   ,,CO.CIR,$COMP
000000                             TRAN$  !MODXPR,PMXPRA   ,,CO.MXP,$COMP
000000                             TRAN$  !PMXPRX,$EXIT    ,,CO.MXP,$COMP
000000                             TRAN$  !MODXSV,PMXSRA   ,,CO.MXS,$COMP
000000                             TRAN$  !MODXAC,PMXACA   ,,CO.MAC,$COMP
000000                             TRAN$  !MODX29,PMXSRA   ,,CO.MX9,$COMP
                          ;;;      TRAN$  !PMXSRV,$EXIT    ,,CO.MXS,$COMP
000000                             TRAN$  !MODSPR,$EXIT,$SNPUR,CO.MSP,$COMP ; PURGE MOD SNA-PROTOCOL
                          ;
                          ; PURGE OPTIONS
                          ;
000000                            STATE$  PUROPT
000000                             TRAN$  &K$ALL, EXIT
000000                             TRAN$  $LAMDA, EXIT
                          ;
                          ; PURGE NODE OPTIONS
                          ;
000000                            STATE$  PURNOD, &P$PUOP
                          ;;;      TRAN$  $EOS,    EXIT,,  RN.NAM, $OPRND
000000                             TRAN$  &K$ALL,  EXIT,,  RN.NAM, $OPRND
000000                             TRAN$  $LAMDA
000000                            STATE$  PNOPT
000000                             TRAN$  !PNHOS,  PNOPT,,         RN.HOS, $OPRND  ;Host node address
000000                             TRAN$  !PNSDV,  PNOPT,,         RN.SDV, $OPRND  ;Service device
000000                             TRAN$  !PNSNV,  PNOPT,,         RN.SNV, $OPRND  ;Service node version
000000                             TRAN$  !PNDPA,  PNOPT,,         RN.DPA, $OPRND  ;Dump address
000000                             TRAN$  !PNDCT,  PNOPT,,         RN.DCT, $OPRND  ;Dump count
000000                             TRAN$  !PNSCI,  PNOPT,,         RN.SCI, $OPRND  ;Service circuit
000000                             TRAN$  !PNHAR,  PNOPT,,         RN.HAR, $OPRND  ;Hardware address
000000                             TRAN$  !PNSPA,  PNOPT,,         RN.SPA, $OPRND  ;Service password
000000                             TRAN$  !PNSLO,  PNOPT,,         RN.SLO, $OPRND  ;Secondary loader file
000000                             TRAN$  !PNTLO,  PNOPT,,         RN.TLO, $OPRND  ;Tertiary loader file
000000                             TRAN$  !PNLOD,  PNOPT,,         RN.LOD, $OPRND  ;Load file
000000                             TRAN$  !PNDIA,  PNOPT,,         RN.DIA, $OPRND  ;Diagnostics file
000000                             TRAN$  !PNDMP,  PNOPT,,         RN.DMP, $OPRND  ;Dump file
000000                             TRAN$  $EOS,    $EXIT,  .RNCHK
                          ;
                          ; Purge Host node address
                          ;
000000                            STATE$  PNHOS
000000                             TRAN$  &K$HOS, $EXIT

                          ;
```

```
   1138 000000                                    TRAN$   &K$KNO
   1139
   1140 000000                                   STATE$   ,&P$PEN
   1141 000000                                    TRAN$   &K$PROS,$EXIT,,WC.PRO,$WILD
   1142                                   ;
   1143                                   ; LOOK FOR ''PROCESS PROCESS-ID''
   1144                                   ;
   1145 000000                                   STATE$   PROID
   1146 000000                                    TRAN$   &K$PRO
   1147
   1148 000000                                   STATE$   ,&P$PID
   1149 000000                                    TRAN$   $RAD50,$EXIT,,.PRNAM
   1818 000000                                          STATE$  ACOS2
   1819 000000                                           TRAN$  &K$COS
   1820 000000                                          STATE$  ,&P$ACOS
   1821 000000                                           TRAN$  $DNUMB, $EXIT,  .EACOS
   1822 000000                                           TRAN$  $LAMDA, $FAIL
   1823                                         ;
   1824                                         ; Executor Broadcast Routing [Timer]
   1825                                         ;
   1826 000000                                          STATE$  EBRT
   1827 000000                                           TRAN$  &K$BRO
   1828 000000                                          STATE$  ,       &P$BROO
   1829 000000                                           TRAN$  &K$TIM, EBRT3
   1830 000000                                           TRAN$  &K$ROU
   1831 000000                                          STATE$  EBRT2,  &P$ROU
   1832 000000                                           TRAN$  &K$TIM, EBRT3
   1833 000000                                           TRAN$  $LAMDA
   1834 000000                                          STATE$  EBRT3,  &P$BRT
   1835 000000                                           TRAN$  $DNUMB, $EXIT,  .EXBRT
   1836 000000                                           TRAN$  $LAMDA, $FAIL
   1837
   1838
   1839                                         ;
   1840                                         ; Executor [Maximum] [Broadcast] Nonrouters
   1841                                         ;
   1842 000000                                          STATE$  EBEA
   1843 000000                                           TRAN$  &K$MAX, EBEA1
   1844 000000                                           TRAN$  &K$BRO, EBEA2
   1845 000000                                           TRAN$  &K$NONR,EBEA3
   1846 000000                                          STATE$  EBEA1,  &P$MAX
   1847 000000                                           TRAN$  &K$BRO, EBEA2
   1848 000000                                           TRAN$  $LAMDA
   1849 000000                                          STATE$  EBEA2
   1850 000000                                           TRAN$  &K$NONR
   1851 000000                                          STATE$  EBEA3,  &P$BEA
   1852 000000                                           TRAN$  $DNUMB, $EXIT,  .EXBEA
   1853 000000                                           TRAN$  $LAMDA, $FAIL
   1854
   1855                                         ;
   1856                                         ; Executor Segment [Buffer] [Size]
   1857                                         ;
   1858 000000                                          STATE$  ESEG
   1859 000000                                           TRAN$  &K$SEG
   1860 000000                                          STATE$
   1861 000000                                           TRAN$  &K$BUF, ESEG2
   1862 000000                                           TRAN$  $LAMDA
   1863 000000                                          STATE$  ESEG2
   1864 000000                                           TRAN$  &K$SIZ, ESEG3
   1865 000000                                           TRAN$  $LAMDA
   1866 000000                                          STATE$  ESEG3,  &P$SEG
   1867 000000                                           TRAN$  $DNUMB, $EXIT,  .EXSEG
   1868 000000                                           TRAN$  $LAMDA, $FAIL
   1869                                         ;
   1870                                         ; EXECUTOR INACTIVITY [TIMER]
   1871                                         ;
   1872 000000                                          STATE$  EINAC
   1873 000000                                           TRAN$  &K$INA
   1874 000000                                          STATE$
```

```
  2427                                            .SBTTL  UTILITY SUBEXPRESSIONS
  2428                                    ;
  2429                                    ; DESTINATION NUMBER (EXTENDED DTE ADDRESS INCLUDING '*')
  2430                                    ;
  2431 000000                                     STATE$  DNUM
  2432 000000                                      TRAN$  &K$NUM
  2433 000000                                     STATE$  ,       &P$DTE
  2434 000000                                      TRAN$  !DNUM0, $EXIT,   .CIDTE
  2435 000000                                     STATE$  DNUM0
  2436 000000                                      TRAN$  $DIGIT, DNUM1
  2437 000000                                      TRAN$  <'*>
  2438 000000                                     STATE$  DNUM1
  2439 000000                                      TRAN$  $DIGIT, DNUM1
  2440 000000                                      TRAN$  <'*>,   DNUM1
  2441 000000                                      TRAN$  $LAMDA, $EXIT
  2442                                    ;
  2443                                    ; DTE ADDRESS
  2444                                    ;
  2445 000000                                     STATE$  DTE
  2446 000000                                      TRAN$  &K$DTE
  2447 000000                                     STATE$  DTENUM, &P$DTE
  2448 000000                                      TRAN$  !DTE0,  $EXIT,   .CIDTE
  2449 000000                                     STATE$  DTE0
  2450 000000                                      TRAN$  $DIGIT,,$BL.MAN
  2451 000000                                     STATE$  DTE1
  2452 000000                                      TRAN$  $DIGIT, DTE1
  2453 000000                                      TRAN$  $LAMDA, $EXIT,$BL.AUT
  2454                                    ;
  2455                                    ; OBJECT NUMBER
  2456                                    ;
  2457 000000                                     STATE$  OBJN
  2458 000000                                      TRAN$  &K$OBJ
  2459 000000                                     STATE$  ,       &P$OBJ
  2460 000000                                      TRAN$  $DNUMB, $EXIT,   .MOOBN
  2461                                    ;
  2462                                    ; OBJECT ID
  2463                                    ;
  2464 000000                                     STATE$  OBJI
  2465 000000                                      TRAN$  &K$OBJ
  2466 000000                                     STATE$  ,       &P$OBJ
  2467 000000                                      TRAN$  $RAD50, $EXIT,   .MOOBJ
  2468                                    ;
  2469                                    ; CALL MASK
  2470                                    ;
  2471 000000                                     STATE$  CLM
  2472 000000                                      TRAN$  &K$CALL
  2473 000000                                     STATE$  ,       &P$CALL
  2474 000000                                      TRAN$  &K$MSK
  2475 000000                                     STATE$  ,       &P$CLM
  2476 000000                                      TRAN$  !HEXSTR,$EXIT,   .MOCLM
  2477                                    ;
  2478                                    ; CALL VALUE
  2479                                    ;
  2480 000000                                     STATE$  CLV
  2481 000000                                      TRAN$  &K$CALL
  2482 000000                                     STATE$  ,       &P$CALL
  2483 000000                                      TRAN$  &K$VAL
```

```
                                                                      B 12
CFEPRS      CREATED BY  MACRO  ON 2-JUL-85 AT 14:24        PAGE 5
MACRO CROSS REFERENCE                                      CREF   04.00
MACRO NAME         REFERENCES

                   #8-1093    #8-1094    #8-1099    #8-1100    #8-1101    #8-1106    #8-1109    #8-1110    #8-1115    #8-1118
                   #8-1121    #8-1126    #8-1129    #8-1132    #8-1138    #8-1141    #8-1146    #8-1149    #9-1156    #9-1157
                   #9-1160    #9-1163    #9-1164    #9-1169    #9-1170    #9-1171    #9-1172    #9-1173    #9-1174    #9-1175
                   #9-1176    #9-1177    #9-1178    #9-1179    #9-1180    #9-1181    #9-1182    #9-1183    #9-1184    #9-1185
                   #9-1190    #9-1192    #9-1193    #9-1195    #9-1196    #9-1198    #9-1203    #9-1205    #9-1206    #9-1208
                   #9-1214    #9-1216    #9-1217    #9-1218    #9-1224    #9-1227    #9-1228    #9-1233    #9-1236    #9-1241
                   #9-1244    #9-1250    #9-1253    #9-1256    #9-1261    #9-1263    #9-1268    #9-1270    #9-1272    #9-1277
                   #9-1279    #9-1281    #9-1286    #9-1288    #9-1290    #9-1295    #9-1297    #9-1299    #9-1304    #9-1306
                   #9-1308    #9-1313    #9-1315    #9-1316    #9-1317    #9-1323    #9-1324    #9-1327    #9-1330    #9-1331
                   #9-1336    #9-1337    #9-1338    #9-1339    #9-1340    #9-1341    #9-1342    #9-1343    #9-1344    #9-1345
                   #9-1346    #9-1347    #9-1348    #9-1349    #9-1350    #9-1355    #9-1358    #9-1359    #9-1360    #9-1365
                   #9-1368    #9-1373    #9-1376    #9-1381    #9-1384    #9-1385    #9-1390    #9-1393    #9-1394    #9-1399
                   #9-1401    #9-1406    #9-1407    #9-1410    #9-1413    #9-1418    #9-1419    #9-1422    #9-1425    #9-1430
                   #9-1433    #9-1436    #9-1437    #9-1440    #9-1441    #9-1442    #9-1443    #9-1444    #9-1445    #9-1446
                   #9-1447    #9-1448    #9-1449    #9-1450    #9-1451    #9-1452    #9-1453    #9-1454    #9-1455    #9-1460
                   #9-1463    #9-1466    #9-1471    #9-1473    #9-1475    #9-1480    #9-1482    #9-1484    #9-1489    #9-1491
                   #9-1493    #9-1498    #9-1500    #9-1502    #9-1507    #9-1509    #9-1511    #9-1518    #9-1521    #9-1522
                   #9-1527    #9-1530    #9-1531    #9-1537    #9-1543    #9-1546    #9-1549    #9-1550    #9-1553    #9-1554
                   #9-1557    #9-1558    #9-1561    #9-1562    #9-1565    #9-1567    #9-1570    #9-1571    #9-1577    #9-1580
                   #9-1585    #9-1588    #10-1595   #10-1598   #10-1601   #10-1602   #10-1607   #10-1608   #10-1609   #10-1610
                   #10-1611   #10-1612   #10-1613   #10-1614   #10-1615   #10-1616   #10-1617   #10-1618   #10-1619   #10-1620
                   #10-1621   #10-1622   #10-1623   #10-1624   #10-1625   #10-1626   #10-1627   #10-1628   #10-1633   #10-1636
                   #10-1641   #10-1644   #10-1649   #10-1652   #10-1654   #10-1655   #10-1656   #10-1662   #10-1665   #10-1668
                   #10-1673   #10-1674   #10-1677   #10-1678   #10-1683   #10-1686   #10-1687   #10-1688   #10-1693   #10-1696
                   #10-1697   #10-1698   #10-1701   #10-1703   #10-1708   #10-1711   #10-1712   #10-1715   #10-1716   #10-1721
                   #10-1724   #10-1727   #10-1732   #10-1735   #10-1738   #10-1743   #10-1746   #10-1749   #10-1754   #10-1757
                   #10-1760   #10-1763   #10-1768   #10-1770   #10-1772   #10-1777   #10-1779   #10-1781   #10-1782   #10-1784
                   #10-1790   #10-1792   #10-1794   #10-1795   #10-1800   #10-1802   #10-1803   #10-1805   #10-1807   #10-1808
                   #10-1814   #10-1816   #10-1817   #10-1819   #10-1821   #10-1822   #10-1827   #10-1829   #10-1830   #10-1832
                   #10-1833   #10-1835   #10-1836   #10-1843   #10-1844   #10-1845   #10-1847   #10-1848   #10-1850   #10-1852
                   #10-1853   #10-1859   #10-1861   #10-1862   #10-1864   #10-1865   #10-1867   #10-1868   #10-1873   #10-1875
                   #10-1876   #10-1878   #10-1879   #10-1884   #10-1886   #10-1887   #10-1889   #10-1890   #10-1895   #10-1897
                   #10-1898   #10-1900   #10-1901   #10-1906   #10-1908   #10-1909   #10-1911   #10-1912   #11-1919   #11-1922
                   #11-1925   #11-1926   #11-1931   #11-1932   #11-1933   #11-1934   #11-1939   #11-1942   #11-1947   #11-1950
                   #11-1951   #11-1956   #11-1959   #11-1960   #11-1965   #11-1968   #11-1969   #11-1970   #11-1975   #11-1978
                   #11-1981   #11-1984   #12-1991   #12-1994   #12-1997   #12-1998   #12-2003   #12-2004   #12-2005   #12-2006
                   #12-2007   #12-2008   #12-2009   #12-2015   #12-2018   #12-2019   #12-2020   #12-2025   #12-2028   #12-2033
                   #12-2036   #12-2041   #12-2044   #12-2045   #12-2048   #12-2049   #12-2052   #12-2057   #12-2058   #12-2061
                   #12-2064   #12-2065   #12-2070   #12-2073   #12-2076   #12-2079   #12-2084   #12-2087   #12-2090   #12-2093
                   #12-2098   #12-2101   #12-2104   #12-2107   #12-2112   #12-2115   #12-2118   #12-2121   #12-2126   #12-2129
                   #12-2132   #12-2135   #13-2142   #13-2143   #13-2144   #13-2145   #13-2146   #13-2147   #13-2152   #13-2153
                   #13-2155   #13-2156   #13-2161   #13-2162   #13-2164   #13-2165   #13-2167   #13-2168   #13-2173   #13-2175
                   #13-2176   #13-2178   #13-2179   #13-2184   #13-2185   #13-2187   #13-2188   #13-2190   #13-2191   #13-2196
                   #13-2197   #13-2199   #13-2201   #13-2202   #13-2204   #13-2205   #13-2206   #13-2207   #13-2208   #13-2209
                   #13-2210   #13-2211   #13-2216   #13-2218   #13-2219   #13-2222   #13-2223   #13-2224   #13-2225   #13-2226
                   #13-2227   #13-2228   #13-2229   #13-2230   #13-2231   #13-2232   #13-2233   #13-2238   #13-2239   #13-2241
                   #13-2243   #13-2244   #13-2247   #13-2248   #13-2249   #13-2250   #13-2251   #13-2257   #13-2258   #13-2260
                   #13-2262   #13-2263   #13-2266   #13-2267   #13-2268   #13-2274   #13-2276   #13-2278   #13-2283   #13-2285
                   #13-2287   #13-2292   #13-2294   #13-2296   #13-2301   #13-2303   #13-2305   #13-2310   #13-2312   #13-2314
                   #13-2320   #13-2322   #13-2324   #13-2329   #13-2331   #13-2333   #13-2338   #13-2340   #13-2342   #13-2347
                   #13-2349   #13-2351   #13-2356   #13-2358   #13-2360   #13-2366   #13-2368   #13-2370   #13-2376   #13-2378
                   #13-2380   #13-2385   #13-2387   #13-2389   #13-2394   #13-2396   #13-2397   #13-2402   #13-2404   #13-2409
```

```
                                                                      C 12
CFEPRS      CREATED BY  MACRO  ON 2-JUL-85 AT 14:24        PAGE 6
```

```
	.TITLE	CFERQB - DYNAMIC BUFFER MANAGEMENT
	.IDENT	/V05.00/
	.ENABL	LC
;
; COPYRIGHT (C) 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;	CFE DYNAMIC BUFFER MANAGEMENT ROUTINES
;
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;

```

```
                        ;****
                        ; INITIALIZE TABLE GENERATION
                        ;****

000002                          ISTAT$  TMPSTB,TMPKTB
                        ;
                        ; SEARCH TABLE FOR DEFINITION
                        ;
000002                          STATE$
000002                            TRAN$   %BUF$DF%,$EXIT,$TMBUF
000002                            TRAN$   %LLC$DF%,$EXIT,$TMLLC
000002                            TRAN$   %DLC$DF%,$EXIT,$TMDLC
000002                            TRAN$   %DDM$DF%,$EXIT,$TMDDM
000002                            TRAN$   %CNT$DF%,$EXIT,$TMCNT
000002                            TRAN$   %SLT$DF%,$EXIT,$TMSLT
000002                            TRAN$   %UNT$DF%,$EXIT,$TMUNT
000002                            TRAN$   %NOD$DF%,$EXIT,$TMNOD
000002                            TRAN$   %DEC$DF%,$EXIT,$TMDEC
000002                            TRAN$   %PAR$DF%,$EXIT,$TMPAR
000002                            TRAN$   %EVT$DF%,$EXIT,$TMEVT
000002                            TRAN$   %LOG$ST%,$EXIT,$TMLOG
000002                            TRAN$   %OBJ$DF%,$EXIT,$TMOBJ
000002                            TRAN$   %FEA$DF%,$EXIT,$TMFEA
000002                            TRAN$   %REM$DF%,$EXIT,$TMREM
000002                            TRAN$   %SER$DF%,$EXIT,$TMSER
000002                            TRAN$   %FIL$DF%,$EXIT,$TMFIL
000002                            TRAN$   %ROU$DF%,$EXIT,$TMROU
000002                            TRAN$   %STA$DF%,$EXIT,$TMSTA
000002                            TRAN$   %PSN$DF%,$EXIT,$TMPSN
000002                            TRAN$   %X3P$DF%,$EXIT,$TMX3P
000002                            TRAN$   %DTE$DF%,$EXIT,$TMDTE
000002                            TRAN$   %CHN$DF%,$EXIT,$TMCHN
000002                            TRAN$   %PVC$DF%,$EXIT,$TMPVC
000002                            TRAN$   %CUG$DF%,$EXIT,$TMCUG
000002                            TRAN$   %DST$DF%,$EXIT,$TMDST
000002                            TRAN$   %DSA$DF%,$EXIT,$TMDSA
000002                            TRAN$   %DSC$DF%,$EXIT,$TMDSC
000002                            TRAN$   %RDT$DF%,$EXIT,$TMRDT
000002                            TRAN$   %X2P$DF%,$EXIT,$TMX2P
000002                            TRAN$   %SVC$DF%,$EXIT,$TMSVC
000002                            TRAN$   %X29$DF%,$EXIT,$TMX29
000002                            TRAN$   %SNA$DF%,$EXIT,$TMSNA
000002                            TRAN$   %SAC$DF%,$EXIT,$TMSAC
000002                            TRAN$   %SAD$DF%,$EXIT,$TMSAD
000002                            TRAN$   %SPU$DF%,$EXIT,$TMSPU
000002                            TRAN$   %END$DF%,$EXIT,,CM.EXI,$CMAND
000002                            TRAN$   $LAMDA,$EXIT

                        ;
                        ; FINAL STATE
                        ;
000002                          STATE$
```

```
                                                .SBTTL  $PRINT - PRINT ASCIZ STRING
                                        ;+
                                        ; *** - $PRINT - PRINT ASCIZ STRING
                                        ;
                                        ; INPUT:
                                        ;       R0 = ADDRESS OF ASCIZ STRING
                                        ;
                                        ; OUTPUT:
                                        ;       STRING PRINTED
                                        ;
                                        ; R0 MODIFIED
                                        ;-
                                        
000000                                  $PRINT::
000000  012767  000000G 000000G                 MOV     #IO.WVB,$TIDPB+Q.IOFN   ; SET FUNCTION CODE
000006  010146                                  MOV     R1,-(SP)                ; SAVE REGISTER
000010  012701  000000G                         MOV     #$TIDPB,R1              ; POINT TO DPB
000014                                  PRINT:
000014  010061  000000G                         MOV     R0,Q.IOPL(R1)           ; STORE ADDRESS OF STRING
000020  105720                          10$:    TSTB    (R0)+                   ; AT END OF STRING ?
000022  001403                                  BEQ     20$                     ; IF EQ, YES
000024  100375                                  BPL     10$                     ; IF PL, NO
000026  005061  000004G                         CLR     Q.IOPL+4(R1)            ; IF MI, NO CARRAIGE CONTROL
000032  166100  000000G                 20$:    SUB     Q.IOPL(R1),R0           ; CALCULATE STRING LENGTH
000036  005300                                  DEC     R0                      ; DON'T COUNT NULL BYTE
000040  010061  000002G                         MOV     R0,Q.IOPL+2(R1)         ; STORE STRING LENGRH
000044                                          DIR$    R1                      ; PRINT STRING
000050  012761  000040  000004G                 MOV     #40,Q.IOPL+4(R1)        ; RESTORE CARRAIGE CONTROL
000056  012601                                  MOV     (SP)+,R1                ; RESTORF REGISTER
000060  000207                                  RETURN
```

$UPBLK - UPDATE POOL BLOCK ALLOCATION

```
    001420  100001                            BPL     44$               ; IF PL, CETAB VALUE IS BIG ENOUGH
    001422  005000                            CLP     R0                ; ELSE BUMP IT UP
    001424  010067  000000G          44$:     MOV     R0,$PBLK          ; SAVE IT
    001430  000406                            BR      47$               ; RETURN SUCCESS
    001432  060001                   45$:     ADD     R0,R1             ; ADD FUDGE FACTOR INTO CALCULATED SIZE
    001434  010 62  000010G                   MOV     R1,C.PRM+10(R2)   ; UPDATE ALLOCATION IN PARTITION TEMPLATE
    001440  052762  000000G 000000G           BIS     #CS.MOD,C.STS(R2) ; INDICATE THAT THE PAR$DF HAS BEEN MODIFIED
    001446  005727                   47$:     TST     (PC)+             ; RETURN SUCCESS
    001450  000261                   50$:     SEC                       ; RETURN FAILURE
    001452  000207                            RETURN

            000001                            .END
```

```
                              ;
                              ; DEFINE THE X.29 PARAMETER BLOCK
                              ;
000036                                .ASECT
              000000                  .=0
000000                                .BLKW   1               ; C.LNK
000002                                .BLKW   1               ; C.STS
000004                        X$9MXC::.BLKW   1               ; MAXIMUM CIRCUITS
000006                        X$9CT:: .BLKW   1               ; COUNTER TIMER
              000010          X$9LEN==.
002054                                .PSECT

                              ;*****
                              ;
                              ; DEFINE THE X.25 LEVEL 3 PARAMETER BLOCK
                              ;
000010                                .ASECT
              000000                  .=0
000000                                .BLKW   1               ; X$3LNK
000002                                .BLKW   1               ; X$3STS
000004                        X$3DFP::.BLKW   1               ; DEFAULT BLOCK SIZE     (21.-1029.)
000006                        X$3MXP::.BLKW   1               ; MAXIMUM BLOCK SIZE     (21.-1029.)
000010                        X$3DW:: .BLKB   1               ; DEFAULT WINDOW SIZE    (1.-127.)
000011                        X$3MW:: .BLKB   1               ; MAXIMUM WINDOW SIZE    (1.-127.)
000012                        X$3CT:: .BLKB   1               ; CALL TIMER VALUE       (0.-255.)
000013                        X$3KT:: .BLKB   1               ; CLEAR TIMER VALUE      (0.-255.)
000014                        X$3RT:: .BLKB   1               ; RESET TIMER VALUE      (0.-255.)
000015                        X$3ST:: .BLKB   1               ; RESTART TIMER VALUE    (0.-255.)
000016                        X$3KM:: .BLKB   1               ; MAXIMUM CLEAR VALUE    (0.-255.)
000017                        X$3RM:: .BLKB   1               ; MAXIMUM RESETS VALUE   (0.-255.)
000020                        X$3SM:: .BLKB   1               ; MAXIMUM RESTARTS VALUE(0.-255.)
              000021          X$3LEN==.                       ; LENGTH OF THE BLOCK
002054                                .PSECT

                              ;****
                              ;
                              ; DEFINE THE DESTINATION PARAMETER BLOCK
                              ;
000021                                .ASECT
              000000                  .=0
000000                                .BLKW   1               ; C.LNK
000002                                .BLKW   1               ; C.STS
000004                        D$STYP::.BLKW   1               ; RAD50 "X25" OR "X29"
000006                        D$SNAM::.BLKW   2               ; RAD50 DESTINATION NAME
000012                        D$SPRI::.BLKB   1               ; LIST INSERTION PRIORITY
000013                        D$SOBJ::.BLKB   1               ; OBJECT NUMBER
000014                        D$STSK::.BLKW   2               ; RAD50 TASK NAME
              000020          D$SLEN==.
002054                                .PSECT
                              ;****
                              ;
                              ; DEFINE THE DSA PARAMETER BLOCK
                              ;
000020                                .ASECT
              000000                  .=0
000000                                .BLKW   1               ; C.LNK
000002                                .BLKW   1               ; C.STS
```

```
   1204                                                   .SBTTL  ERROR MESSAGES
   1205                                           ;
   1206                                           ; ERROR MESSAGES
   1207                                           ;
   1208                                                   .NLIST  BEX
   1209                                                   .ENABL  LC
   1210                 000000                            EX$WAR = 0
   1211                 000001                            EX$SUC = 1
   1212                 000002                            EX$ERR = 2
   1213                 000004                            EX$SEV = 4
   1214 002054             002        103        106     ERRSYN::.ASCIZ  <EX$ERR>/CFE -- Syntax error/
   1215 002101             004        103        106     ERRCON::.ASCIZ  <EX$SEV>/CFE -- Consistency error/
   1216 002133             002        103        106     ERRILC::.ASCIZ  <EX$ERR>/CFE -- Illegal command/
   1217 002163             002        103        106     ERRGCL::.ASCIZ  <EX$ERR>/CFE -- Command line error/
   1218 002216             002        103        106     ERRILN::.ASCIZ  <EX$ERR>/CFE -- Invalid number/
   1219 002245             004        103        106     ERRFIL::.ASCIZ  <EX$SEV>/CFE -- Error in opening file/
   1220 002303             004        103        106     ERROUT::.ASCIZ  <EX$SEV>/CFE -- Error writing file/
   1221 002336             004        103        106     ERRALL::.ASCIZ  <EX$SEV>/CFE -- Insufficient buffer space/
   1222 002400             002        103        106     ERRPRC::.ASCIZ  <EX$ERR>/CFE -- Process not in configuration/
   1223 002445             002        103        106     ERRLIS::.ASCIZ  <EX$ERR>/CFE -- Line not in configuration/
   1224 002507             002        103        106     ERRCIS::.ASCIZ  <EX$ERR>/CFE -- Circuit not in configuration/
   1225 002554             002        103        106     ERRNOD::.ASCIZ  <EX$ERR>/CFE -- Node not in configuration/
   1226 002616             002        103        106     ERRPNA::.ASCIZ  <EX$ERR>/CFE -- Parameter not applicable/
   1227                                                   .LIST   BEX
   1228                                                   .EVEN
   1229
   1230
   1231                 000001                            .END
```

```
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE             REFERENCES

SV$DTE  = 002000  G       #12-1194
SV$PSZ  = 000400  G       #12-1192
SV$WSZ  = 001000  G       #12-1193
SW.ENA  = 000004  G       #11-731
SW.FUL  = 000010  G       #11-732
SW.INC  = 002000  G       #11-739
SW.LOA  = 000002  G       #11-730
SW.LON  = 000200  G       #11-736
SW.OUT  = 001000  G       #11-738
SW.PUR  = 000400  G       #11-737
SW.RLU  = 000020  G       #11-733
SW.SDI  = 010000  G       #11-741     11-743
SW.SEN  = 004000  G       #11-740     11-743
SW.SLT  = 014000  G       #11-743
SW.SMC  = 000040  G       #11-734
SW.TOP  = 000001  G       #11-729
SW.VFY  = 000100  G       #11-735
S$LBRA    000022  G       #9-362
S$LCNT    000010  G       #9-355
S$LCTT    000012  G       #9-357
S$LFLG    000006  G       #9-354
S$LHTM    000016  G       #9-360
S$LLEN  = 000024  G       #9-363
S$LNAM    000004  G       #9-353
S$LRPR    000020  G       #9-361
S$LTYP    000014  G       #9-358
S$LUNT    000011  G       #9-356
S$TADD    000005  G       #9-426
S$TAPR    000010  G       #9-429
S$TCST    000007  G       #9-428
S$TFLG    000006  G       #9-427
S$THTM    000012  G       #9-431
S$TLEN  = 000016  G       #9-433
S$TSTA    000004  G       #9-425
S$VDTE    000022  G       #9-566
S$VFLG    000014  G       #9-563
S$VLEN  = 000036  G       #9-568
S$VNAM    000004  G       #9-560
S$VOWN    000034  G       #9-567
S$VPSZ    000010  G       #9-561
S$VRCL    000016  G       #9-564
S$VTIM    000020  G       #9-565
S$VWSZ    000012  G       #9-562
S.BFHD  = 000020            6-92      6-92
S.ETEF  = 000002            6-95
S.FDB   = 000140            6-95      6-95      6-104     6-104     6-110     6-110
S.FNAM  = 000006            6-95      6-95      6-115     6-115
S.FNB   = 000036            6-95      6-115
S.FTYP  = 000002            6-95      6-95      6-115     6-115
UN$CST  = 001000  G       #12-1197
UN$DPR  = 002000  G       #12-1198
UN$PCH  = 004000  G       #12-1199
```

```
                                          .SBTTL  $CFINI - INITIALIZE END OF TASK BUFFERS
                                  ;+
                                  ; *** - $CFINI - INITIALIZE END OF TASK BUFFERSS
                                  ;
                                  ; THE CETAB FILE-SPEC IS INPUT BY THE USER AND THE FILE IS OPENED AND
                                  ; READ INTO THE END-OF-TASK TEMPLATES. A NEW FILE IS OPENED FOR OUTPUT.
                                  ;
                                  ; INPUT:
                                  ;         NONE.
                                  ;
                                  ; OUTPUT:
                                  ;         C-BIT = SUCCESS/FAILURE
                                  ;
                                  ;-

000024                            $CFINI::
000024  012767  000000G 000000G           MOV     #EX$SUC,$EXSTS  ; ASSUME SUCESSFUL EXECUTION OF CFE
000032  004767  000000G                   CALL    $XXINI          ; INITIALIZE END OF TASK BUFFERS
000036                                    FINIT$                  ; INITIALIZE FILE CONTROL SYSTEM
000042  016746  000000G           5$:     MOV     $CLPMT,-(SP)    ; SAVE CURRENT PROMPT STRING
000046  012767  000000' 000000G           MOV     #PRMPT,$CLPMT   ; PROMPT FOR FILENAME
000054  016700  000000G                   MOV     $CLIOS,R0       ; POINT R0 AT I/O STATUS BLOCK
000060  042767  000000G 000000G           BIC     #CL.CMT,$CLFLG  ; ALLOW VERSION NUMBER
000066                                    GCL$                    ; GET A COMMAND LINE
000072  052767  000000G 000000G           BIS     #CL.CMT,$CLFLG  ; FILTER COMMENTS
000100  012667  000000G                   MOV     (SP)+,$CLPMT    ; RESTORE PREVIOUS PROMPT
000104  103010                            BCC     10$             ; IF CC, OKAY
000106  122710  000000G                   CMPB    #IE.EOF,(R0)    ; JUST AN END-OF-FILE ?
000112  001427                            BEQ     30$             ; IF EQ, YES - SKIP THE MESSAGE
000114                                    ERROR$  #ERRGCL         ; ELSE, COMMAND LINE ERROR
000124  000746                            BR      5$              ; AND TRY AGAIN
000126  016002  000002            10$:    MOV     2(R0),R2        ; GET LENGTH OF FILE-SPEC
000132  012701  000000G                   MOV     #$BUFF,R1       ; POINT R1 AT START OF FILE-SPEC
000136  004767  000000G                   CALL    $IOPEN          ; TRY TO OPEN INPUT FILE
000142  103407                            BCS     20$             ; IF CS, ERROR OPENING FILE
000144  004767  000000G                   CALL    $OPEN           ; TRY TO OPEN FILE FOR OUTPUT
000150  103012                            BCC     40$             ; IF CC, FILE WAS OPENED
000152                                    CLOSE$  #$IFDB          ; CLOSE THE INPUT FILE
000162                            20$:    ERROR$  #ERRFIL         ; ELSE, ERROR IN OPENING FILE
000172  000261                    30$:    SEC                     ; SHOW FAILURE
000174  000426                            BR      50$             ; AND LEAVE
000176  012700  000000G           40$:    MOV     #$IFDB,R0       ; POINT AT INPUT FILE FDB
000202  004767  000000G                   CALL    .MARK           ; MARK POSITION OF TOP OF FILE
000206  010167  000000G                   MOV     R1,$HIVBN       ; SAVE HIGH-ORDER BITS OF VBN
000212  010267  000000G                   MOV     R2,$LOVBN       ; SAVE LOW-ORDER BITS OF VBN
000216  010367  000000G                   MOV     R3,$BYTE        ; SAVE BYTE ADDRESS
000222  004767  000000G                   CALL    $TMSTB          ; SET UP THE END OF TASK TEMPLATES
000226  103011                            BCC     50$             ; IF CC, ALL OKAY
000230                                    CLOSE$  #$IFDB          ; CLOSE THE INPUT FILE
000240                                    DELET$  #$OFDB          ; DELETE THE OUTPUT FILE
000250  000261                            SEC                     ; MAKE SURE THE C-BIT IS SET !
000252  000207                    50$:    RETURN


        000001                            .END
```

```
                        .TITLE  CFEOPN - FILE I/O ROUTINES
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - INPUT AND OUTPUT FILE OPEN AND I/O ROUTINES
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  10-FEB-78
                ;       VERSION 2.0 RELEASE
                ;
                ; 2.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
     83                                                         .SBTTL  LOCAL DATA
     84                                                         .NLIST  BEX
     85
     86                                                 ;****
     87                                                 ; LOCAL DATA
     88                                                 ;****
     89
     90                                                         .ENABL  LC
     91                 000002                                  EX$ERR = 2
     92 000000             002         103         106  ERROBJ: .ASCIZ  <EX$ERR>/CFE -- Object not in configuration/
     93 000044             002         103         106  ERRCTY: .ASCIZ  <EX$ERR>/CFE -- Illegal circuit type/
     94 000101             002         103         106  ERRNSC: .ASCIZ  <EX$ERR>/CFE -- Circuit not in configuration/
     95                                                         .EVEN
     96                                                         .DSABL  LC
     97
     98                                                 ;
     99                                                 ; DEFINE THE DISPATCH TABLE
    100                                                 ;
    101 000146                                          DSPTBL:
    102 000146  000000G                                         .WORD   $PUREM          ; ''PURGE REMOTE'' REQUEST
    103 000150  177777                                          .WORD   -1              ; ''PURGE PROCESS'' ???
    104 000152  177777                                          .WORD   -1              ; ''PURGE EXECUTOR'' ???
    105 000154  177777                                          .WORD   -1              ; ''PURGE SYSTEM'' ???
    106 000156  000226'                                         .WORD   PUROBJ          ; ''PURGE OBJECT'' REQUEST
    107 000160  000734'                                         .WORD   PURLIN          ; ''PURGE LINE''
    108 000162  000360'                                         .WORD   PURLOG          ; ''PURGE LOGGING'' REQUEST
    109 000164  000756'                                         .WORD   PURCIR          ; ''PURGE CIRCUIT'' REQUEST
    110 000166  000756'                                         .WORD   PURCIR          ; ''PURGE KNOWN CIRCUITS'' REQUEST
    111 000170  000000G                                         .WORD   $PUXPR          ; ''PURGE MODULE X25-PROTOCOL'' REQUEST
    112 000172  000000G                                         .WORD   $PUXSV          ; ''PURGE MODULE X25-SERVER'' REQUEST
    113 000174  000000G                                         .WORD   $PUXAC          ; ''PURGE MODULE X25-ACCESS'' REQUEST
    114 000176  000000G                                         .WORD   $PUXS9          ; ''PURGE MODULE X29-SERVER'' REQUEST
    115 000200  000000G                                         .WORD   $PUSNA          ; ''PURGE MODULE SNA-PROTOCOL'' REQUEST
    116
    117                                                 ;
    118                                                 ; DEFINE SINK TYPE TABLE
    119                                                 ;
    120 000202                                          SNKTBL:
    121 000202  000001                                          .WORD   FF.CON          ; CONSOLE SINK TYPE
    122 000204  000002                                          .WORD   FF.FIL          ; FILE SINK TYPE
    123 000206  000004                                          .WORD   FF.MON          ; MONITOR SINK TYPE
    124 000210  000000                                          .WORD   0
    125
    126 000212                                          TEMP:   .BLKW   1               ; TEMPORARY STORAGE
    127                                                         .LIST   BEX
```

```
CFEPRG     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57      PAGE 2

MACRO CROSS REFERENCE                                    CREF    04.00

MACRO NAME          REFERENCES

ERROR$              #4-68         8-212        11-397       11-399
FLTDF$              #4-79         4-81
PRINT               #4-63
RESRG               #4-79
SAVRG               #4-79
```

```
**FILE**ID**CFEPRS
```

```
   489                                  ; Purge [Service] device
   490                                  ;
   491 000000                                   STATE$  PNSDV
   492 000000                                    TRAN$  &K$DEV, $EXIT
   493 000000                                    TRAN$  &K$SER
   494 000000                                   STATE$   ,&P$SERO
   495 000000                                    TRAN$  &K$DEV, $EXIT
   496
   497                                  ;
   498                                  ; Purge Service node version
   499                                  ;
   500 000000                                   STATE$  PNSNV
   501 000000                                    TRAN$  &K$VRSN,$EXIT
   502 000000                                    TRAN$  &K$SER
   503 000000                                   STATE$   ,&P$SERO
   504 000000                                    TRAN$  &K$VRSN,$EXIT
   505 000000                                    TRAN$  &K$NOD
   506 000000                                   STATE$
   507 000000                                    TRAN$  &K$VRSN,$EXIT
   508
   509                                  ;
   510                                  ; Purge Dump address
   511                                  ;
   512 000000                                   STATE$  PNDPA
   513 000000                                    TRAN$  &K$DMP
   514 000000                                   STATE$   ,         &P$DMPO
   515 000000                                    TRAN$  &K$ADD, $EXIT
   516
   517                                  ;
   518                                  ; Purge Dump Count
   519                                  ;
   520 000000                                   STATE$  PNDCT
   521 000000                                    TRAN$  &K$DMP
   522 000000                                   STATE$   ,&P$DMPO
   523 000000                                    TRAN$  &K$COUN,$EXIT
   524                                  ;
   525                                  ; Purge [Service] circuit
   526                                  ;
   527 000000                                   STATE$  PNSCI
   528 000000                                    TRAN$  &K$CIR, $EXIT
   529 000000                                    TRAN$  &K$SER
   530 000000                                   STATE$   ,&P$SERO
   531 000000                                    TRAN$  &K$CIR, $EXIT
   532                                  ;
   533                                  ; Purge Hardware [address]
   534                                  ;
   535 000000                                   STATE$  PNHAR
   536 000000                                    TRAN$  &K$HRD
   537 000000                                   STATE$
   538 000000                                    TRAN$  &K$ADD, $EXIT
   539 000000                                    TRAN$  $LAMDA, $EXIT
   540
   541                                  ;
   542                                  ; Purge [Service] password
   543                                  ;
   544 000000                                   STATE$  PNSPA
   545 000000                                    TRAN$  &K$SER, PNSPA2
```

```
                        .SBTTL  DEFINE CIRCUIT COMMAND
               ;
               ; DEFINE CIRCUIT COMMAND
               ;
000000                  STATE$  DEFCIR
000000                   TRAN$  !CIRIT, CIRC,,   CO.CIR, $COMP
000000                   TRAN$  !KNCIR,,,        CO.KNC, $COMP

000000                  STATE$  CIRC,&P$DECI
000000                   TRAN$  !CIROPT

000000                  STATE$  CIRO
000000                   TRAN$  !CIROPT,CIRO
000000                   TRAN$  $LAMDA, $EXIT
               ;
               ; CIRCUIT OPTIONS
               ;
000000                  STATE$  CIROPT
000000                   TRAN$  !CSTATE,$EXIT,, OC.STA, $OPRND
000000                   TRAN$  !COST,   $EXIT,, OC.CST, $OPRND
000000                   TRAN$  !TRIB,   $EXIT,, OC.TRI, $OPRND
000000                   TRAN$  !APR,    $EXIT,, OC.APR, $OPRND
000000                   TRAN$  !CHN,    $EXIT,, OC.CHN, $OPRND
000000                   TRAN$  !CT,     $EXIT,, OC.CTM, $OPRND
000000                   TRAN$  !DTE,    $EXIT,, OC.DTE, $OPRND
000000                   TRAN$  !HTM,    $EXIT,, OC.HTM, $OPRND
000000                   TRAN$  !MBL,    $EXIT,, OC.MBL, $OPRND
000000                   TRAN$  !MRC,    $EXIT,, OC.MRC, $OPRND
000000                   TRAN$  !MWIL,   $EXIT,, OC.MWI, $OPRND
000000                   TRAN$  !RTM,    $EXIT,, OC.RTM, $OPRND
000000                   TRAN$  !USG,    $EXIT,, OC.USG, $OPRND
000000                   TRAN$  !DNUM,   $EXIT,, OC.NUM, $OPRND
000000                   TRAN$  !CIMXR, $EXIT,, OC.MXR, $OPRND
000000                   TRAN$  !CIRPR, $EXIT,, OC.RPR, $OPRND
000000                   TRAN$  !CISER, $EXIT
               ;
               ; CIRCUIT MAXIMUM [BROADCAST] ROUTERS
               ;
000000                  STATE$  CIMXR
000000                   TRAN$  &K$MAX
000000                  STATE$  ,&P$MAX
000000                   TRAN$  &K$ROUT,CIMXR3
000000                   TRAN$  &K$BRO
000000                  STATE$  CIMXR2
000000                   TRAN$  &K$ROUT,CIMXR3
000000                   TRAN$  $LAMDA
000000                  STATE$  CIMXR3, &P$MXR
000000                   TRAN$  $DNUMB, $EXIT,  .CIMXR
               ;
               ; CIRCUIT ROUTER PRIORITY
               ;
000000                  STATE$  CIRPR
000000                   TRAN$  &K$ROUT
000000                  STATE$
000000                   TRAN$  &K$PRI, CIRPR2
000000                   TRAN$  $LAMDA
000000                  STATE$  CIRPR2, &P$RPR
```

```
   1875 000000                                    TRAN$  &K$TIM,EINA2
   1876 000000                                    TRAN$  $LAMDA
   1877 000000                                   STATE$  EINA2,  &P$INA
   1878 000000                                    TRAN$  $DNUMB, $EXIT,  .EXINA
   1879 000000                                    TRAN$  $LAMDA, $FAIL
   1880                                  ;
   1881                                  ; EXECUTOR INCOMING [TIMER]
   1882                                  ;
   1883 000000                                   STATE$  EINCT
   1884 000000                                    TRAN$  &K$INC
   1885 000000                                   STATE$
   1886 000000                                    TRAN$  &K$TIM,EINC2
   1887 000000                                    TRAN$  $LAMDA
   1888 000000                                   STATE$  EINC2,  &P$INC
   1889 000000                                    TRAN$  $DNUMB, $EXIT,  .EXINT
   1890 000000                                    TRAN$  $LAMDA, $FAIL
   1891                                  ;
   1892                                  ; EXECUTOR OUTGOING [TIMER]
   1893                                  ;
   1894 000000                                   STATE$  EOUTT
   1895 000000                                    TRAN$  &K$OUT
   1896 000000                                   STATE$
   1897 000000                                    TRAN$  &K$TIM,EOUT2
   1898 000000                                    TRAN$  $LAMDA
   1899 000000                                   STATE$  EOUT2,  &P$OUT
   1900 000000                                    TRAN$  $DNUMB, $EXIT,  .EXOUT
   1901 000000                                    TRAN$  $LAMDA, $FAIL
   1902                                  ;
   1903                                  ; EXECUTOR RETRANSMIT [FACTOR]
   1904                                  ;
   1905 000000                                   STATE$  ERETF
   1906 000000                                    TRAN$  &K$RTR
   1907 000000                                   STATE$
   1908 000000                                    TRAN$  &K$FAC,ERET2
   1909 000000                                    TRAN$  $LAMDA
   1910 000000                                   STATE$  ERET2,  &P$RETF
   1911 000000                                    TRAN$  $DNUMB, $EXIT,  .EXRET
   1912 000000                                    TRAN$  $LAMDA, $FAIL
```

```
  2484 000000                                   STATE$  ,          &P$CLM
  2485 000000                                    TRAN$  !HEXSTR,$EXIT,  .MOCLV
  2486                                  ;
  2487                                  ; HEXADECIAL STRING FOR CALL MASK AND VALUE
  2488                                  ;
  2489 000000                                   STATE$  HEXSTR
  2490 000000                                    TRAN$  $LAMDA,,         $BL.MAN
  2491 000000                                   STATE$
  2492 000000                                    TRAN$  !HEXDIG,HEXST2
  2493 000000                                    TRAN$  $LAMDA,,         .BL.AU3 ; RESTORE AUTO BLANKS AND FAIL
  2494 000000                                   STATE$  HEXST2
  2495 000000                                    TRAN$  !HEXDIG,HEXST2
  2496 000000                                    TRAN$  $LAMDA, $EXIT,   $BL.AUT
  2497                                  ;
  2498                                  ; MATCH A HEXADECIMAL DIGIT
  2499                                  ;
  2500 000000                                   STATE$  HEXDIG
  2501 000000                                    TRAN$  $DIGIT, $EXIT
  2502 000000                                    TRAN$  'A,     $EXIT
  2503 000000                                    TRAN$  'B,     $EXIT
  2504 000000                                    TRAN$  'C,     $EXIT
  2505 000000                                    TRAN$  'D,     $EXIT
  2506 000000                                    TRAN$  'E,     $EXIT
  2507 000000                                    TRAN$  'F,     $EXIT
  2508                                  ;
  2509                                  ;LIST INSERTION PRIORITY FOR DESTINATIONS
  2510                                  ;
  2511 000000                                   STATE$  DPRI
  2512 000000                                    TRAN$  &K$PRI
  2513 000000                                   STATE$  ,          &P$LPRI
  2514 000000                                    TRAN$  $DNUMB, $EXIT,   .MOPRI
  2515                                  ;
  2516                                  ; SUBADDRESS RANGE FOR DESTINATIONS
  2517                                  ;
  2518 000000                                   STATE$  DSUB
  2519 000000                                    TRAN$  &K$SUBA
  2520 000000                                   STATE$  ,          &P$SUBA
  2521 000000                                    TRAN$  $DNUMB,,         .MODS1
  2522 000000                                   STATE$
  2523 000000                                    TRAN$  <'->,   DSUB2
  2524 000000                                    TRAN$  $LAMDA, $EXIT
  2525 000000                                   STATE$  DSUB2
  2526 000000                                    TRAN$  $DNUMB, $EXIT,   .MODS2
  2527
  2528                                  ;
  2529                                  ; SUBEXPRESSION TO MATCH MODULE X25-PROTOCOL
  2530                                  ;
  2531 000000                                   STATE$  MODXPR
  2532 000000                                    TRAN$  !MX25
  2533 000000                                   STATE$
  2534 000000                                    TRAN$  &K$PRT, BLAUT
  2535 000000                                    TRAN$  $LAMDA,,         .BL.AU3
  2536                                  ;
  2537                                  ; SUBEXPRESSION TO MATCH MODULE X25-SERVER
  2538                                  ;
  2539 000000                                   STATE$  MODXSV
  2540 000000                                    TRAN$  !MX25
```

```
CFEPRS     CREATED BY  MACRO  ON 2-JUL-85 AT 14:24        PAGE 6
MACRO CROSS REFERENCE                                     CREF   04.00
MACRO NAME        REFERENCES

                  #13-2411   #13-2416   #13-2418   #13-2420   #13-2422   #13-2423   #13-2424   #14-2432   #14-2434   #14-2436
                  #14-2437   #14-2439   #14-2440   #14-2441   #14-2446   #14-2448   #14-2450   #14-2452   #14-2453   #14-2458
                  #14-2460   #14-2465   #14-2467   #14-2472   #14-2474   #14-2476   #14-2481   #14-2483   #14-2485   #14-2490
                  #14-2492   #14-2493   #14-2495   #14-2496   #14-2501   #14-2502   #14-2503   #14-2504   #14-2505   #14-2506
                  #14-2507   #14-2512   #14-2514   #14-2519   #14-2521   #14-2523   #14-2524   #14-2526   #14-2532   #14-2534
                  #14-2535   #14-2540   #14-2542   #14-2543   #14-2548   #14-2550   #14-2551   #14-2556   #14-2558   #14-2559
                  #14-2564   #14-2569   #14-2570   #14-2572   #14-2573   #14-2575   #14-2577   #14-2579   #14-2584   #14-2585
                  #14-2587   #14-2588   #14-2590   #14-2592   #14-2594   #14-2599   #14-2601   #14-2603   #14-2620   #14-2621
                  #14-2624   #14-2627   #14-2630   #14-2633   #14-2634   #15-2643
```

```
**FILE**ID**CFEROT
```

```
    46                                          .SBTTL  MACRO CALLS
    47                                  ;
    48                                  ; MACRO CALLS
    49                                  ;
    50                                          .MCALL  SAVRG,RESRG,EXTK$S
    51
    52                                  ;
    53                                  ; GLOBAL SYMBOL DEFINITION
    54                                  ;
    55          000020                          .EXINC  == 20           ; NUMBER OF 100 BYTE BLOCKS TO EXTEND BY
    56
```

```
                                        .SBTTL  $TMSTB - SET UP CETAB TEMPLATE BLOCKS
                                ;+
                                ; *** - $TMSTB - SET UP CETAB TEMPLATE BLOCKS
                                ;
                                ; INPUT:
                                ;         NONE.
                                ;
                                ; OUTPUT:
                                ;         C-BIT = SUCCESS/FAILURE
                                ;         THE CETAB FILE IS READ INTO THE END OF TASK SPACE AS BINARY INFO.
                                ;
                                ;-

000002                          $TMSTB::
000002  005067  000000G                 CLR     $CMAND            ; CLEAR COMMAND CODE
000006  005067  177766                  CLR     $ERROR            ; CLEAR ERROR INDICATOR
000012  022767  000000G 000000G 10$:    CMP     #CM.EXI,$CMAND    ; TIME TO EXIT ?
000020  001450                          BEQ     30$               ; IF EQ, YES
000022  004767  000000G                 CALL    $RDATA            ; READ A RECORD
000026  103445                          BCS     30$               ; IF CS, END OF FILE
000030  005703                          TST     R3                ; ZERO LENGTH RECORD ?
000032  001767                          BEQ     10$               ; IF EQ, YES
000034  005001                          CLR     R1                ; TPARS INITIALIZATION - EXACT MATCH
000036  012702  000000'                 MOV     #TMPKTB,R2        ; KEYWORD TABLE
000042  012704  000000G                 MOV     #$BUFI,R4         ; ADDRESS OF RECORD
000046  012705  000000'                 MOV     #TMPSTB,R5        ; STATE TABLE
000052  004767  000000G                 CALL    .TPARS            ; PARSE THE RECORD
000056  103404                          BCS     15$               ; IF CS, ERROR
000060  005767  177714                  TST     $ERROR            ; SYNTAX ERROR ?
000064  001752                          BEQ     10$               ; IF EQ, NO .. READ NEXT RECORD
000066  100404                          BMI     20$               ; IF MI, MESSAGE PRINTED
000070                          15$:    ERROR$  #ERRSYN           ; ELSE, SYNTAX ERROR
                                ;
                                ; INSERT ERROR FLAG IN OFFENDING LINE AND PRINT IT
                                ;
000100  010400                  20$:    MOV     R4,R0             ; COPY ADDRESS OF OFFENDING SYMBOL
000102  105720                  22$:    TSTB    (R0)+             ; LOCATE END OF STRING
000104  001376                          BNE     22$               ; IF NE, MORE
000106  114060  000002          24$:    MOVB    -(R0),2(R0)       ; SLIDE EVERYTHING DOWN 2 BYTES
000112  020004                          CMP     R0,R4             ; HAVE WE REACHED START OF ERROR ?
000114  001374                          BNE     24$               ; IF NE, NO
000116  112720  000076                  MOVB    #RBRKT,(R0)+      ; INSERT ERROR FLAG
000122  112720  000076                  MOVB    #RBRKT,(R0)+      ; ...
000126                                  EPRINT  #$BUFI            ; PRINT THE BAD LINE
000136  000261                          SEC                       ; SHOW FATAL ERROR
000140  000401                          BR      40$               ; AND LEAVE
000142  000241                  30$:    CLC                       ; SUCCESS !!!
000144  000207                  40$:    RETURN
```

```
                                        .SBTTL  $CFERR - SET UP EXIT STATUS AND PRINT ASCIZ ERROR MESSAGE
                                        .SBTTL  $CFER1 - PRINT ASCIZ STRING ON ERROR LUN
                                ;+
                                ; *** - $CFERR - PRINT ASCIZ STRING AND SET UP EXIT STATUS
                                ; *** - $CFER1 - JUST PRINT STRING
                                ;
                                ; INPUT:
                                ;       R0 = POINTER TO EXIT STATUS (BYTE) FOLLOWED BY ERROR MESSAGE ($CFERR)
                                ;       R0 = ADDRESS OF ASCIZ STRING ($CFER1)
                                ;
                                ; OUTPUT:
                                ;       STRING PRINTED (AND EXIT STATUS SET UP)
                                ;
                                ; R0 MODIFIED
                                ;-

                                        .ENABL  LSB
000062                          $CFERR::
000062  022767  000000G 000000G         CMP     #EX$SUC,$EXSTS          ; ANY ERRORS YET ?
000070  001403                          BEQ     3$                      ; IF EQ, NO - SET STATUS
000072  121067  000000G                 CMPB    (R0),$EXSTS             ; IS ERROR WORSE THAN LAST ?
000076  101402                          BLOS    5$                      ; IF LOS, NO - DON'T SET STATUS
000100  111067  000000G         3$:     MOVB    (R0),$EXSTS             ; STORE NEW EXIT STATUS
000104  105720                  5$:     TSTB    (R0)+                   ; SKIP STATUS FIELD OF MESSAGE
000106  010146                  $CFER1::MOV     R1,-(SP)                ; SAVE REGISTER
000110  012701  000000G                 MOV     #$ERDPB,R1              ; POINT TO DPB
000114  000737                          BR      PRINT                   ; PRINT STRING

000116  010067  000000C                 MOV     R0,$ERDPB+Q.IOPL        ; STORE ADDRESS OF ERROR MESSAGE
000122  105720                  10$:    TSTB    (R0)+                   ; AT END OF STRING ?
000124  001403                          BEQ     20$                     ; IF EQ, YES
000126  100375                          BPL     10$                     ; IF PL, NO
000130  005067  000000C                 CLR     $ERDPB+Q.IOPL+4         ; ELSE CLEAR CARRAIGE CONTROL
000134  166700  000000C         20$:    SUB     $ERDPB+Q.IOPL,R0        ; CALCULATE STRING LENGTH
000140  005300                          DEC     R0                      ; DON'T COUNT NULL BYTE
000142  010067  000000C                 MOV     R0,$ERDPB+Q.IOPL+2      ; STORE STRING LENGTH
000146                                  DIR$    #$ERDPB                 ; PRINT STRING
000154  012727  000040  000000C         MOV     #40,#$ERDPB+Q.IOPL+4    ; RESTORE CARRAIGE CONTROL
000162  000207                          RETURN
                                        .DSABL  LSB
```

```
[illegible] 000022        D$DTEL   000031        FF.MOD=  000100        LP$ENB=  004000        L$NRPK   000056
[illegible]K= 000102      D$FLEN   000032        FF.MON=  000004        LP$PCT=  001400        L$NRRE   000105
CL$MFL=  000010           D$FLNX=  000034  G     FF.MSK=  000077        LP$TMR=  100000        L$NRST   000107
CL$SFL=  000004           D$GLEN   000032        FF.PRT=  000200        LP$UP =  010000        L$NTBY   000052
CL$TYP=  000001           D$GVBL   000032        FF.QUL=  000370        LP$WTD=  020000        L$NTCA   000070
CL.MU1=  000001           D$LNK    000000        FF.REM=  100000        LP$WTS=  040000        L$NTFS   000102
CL.MU2=  000002           D$NAM    000004        FL.MFL=  ******  GX    LS$GDN=  000004        L$NTPK   000062
CL.RES=  177774           D$NOD    000010        FL.MTP=  ******  GX    LS$OFF=  000000        L$OMST   000002
CNTRL    000404R          D$OBJ    000003        FL.MUX=  ******  GX    LS$STR=  000002        L$PCS    000112
CS.DDM=  ******  GX       D$PAL    000026        FL.TRI=  ******  GX    LS$UP =  000003        L$PLS    000110
CS.DEC=  ******  GX       D$PRI    000002        F.ADD    000016        LS$WT =  000001        L$QCNT   000122
CS.DLC=  ******  GX       D$PRL    000024        F.CEV    000030        LS.DLS=  100000        L$QUEH   000124
CS.LLC=  ******  GX       D$SHI    000020        F.CIR    000024        LS.FCC=  000004        L$QUET   000130
CS.MOD=  ******  GX       D$SLO    000016        F.CLS    000002        LS.FC0=  000001        L$RTRY   000007
CS.PAR=  ******  GX       D$USL    000025        F.EVT    000004        LS.FC1=  000002        L$SLN    000004
CS.ROU=  ******  GX       D$VBL    000032        F.FLG    000014        LS.ILS=  100000        L$ST     000005
CS.SLT=  ******  GX       ER$ABM=  000010        F.LEN    000016        LS.MAK=  000020        L$TCLZ   000044
CS.STA=  ******  GX       ER$ABO=  000046        F.LENX=  000020  G     LS.MNK=  000040        L$TIM    000006
CVTBL    000414R          ER$ABT=  000011        F.LIN    000020        LS.RES=  000360        L$TRI =  ******  GX
CVTC     000316R          ER$ACC=  000042        F.LNK    000000        LS.RSV=  000300        L$UNT =  ******  GX
CV$MSK=  000003           ER$CDI=  000052        F.MOD    000022        LT.CCA=  000020        L.ACC    000146
CV$31 =  000001           ER$COM=  000047        F.REM    000026        LT.DIR=  000010        L.CHN    000124
CV$32 =  000000           ER$FMT=  000005        F.SEV    000004        LT.LCL=  000001        L.CINT   000142
CV$40 =  000002           ER$MLB=  000006        GF$BUG=  000001        LT.LPL=  000002        L.CSTA   000037
C$DTE    000012           ER$NNF=  000012        G$CUG    000010        LT.NOT=  000040        L.CTR    000074
C$FLG    000015           ER$NOD=  000002        G$DTE    000012        LT.RSU=  000200        L.CXMT   000134
C$GNAM   000016           ER$NSL=  000013        G$FLG    000014        LT.SLI=  000004        L.DCR    000100
C$LCN    000010           ER$NSR=  000003        G$GNAM   000016        LT.TDA=  000100        L.FLAG   000014
C$LEN    000016           ER$RES=  000001        G$LEN    000016        L$ACHN   000012        L.ILSQ   000052
C$LENX=  000020  G        ER$STA=  000051        G$LENX=  000020  G     L$APVC   000014        L.ILTT   000066
C$LNK    000000           ER$UOB=  000004        G$LNK    000000        L$ASVC   000010        L.INTQ   000136
C$NAM    000002           EXTCHR   000232R       G$NAM    000002        L$AUC    000042        L.LDA    000032
C$NML    000002           EX$SUC=  ******  GX    G$NML    000015        L$CHLS   000034        L.LIA    000034
C$PORT   000014           E$NBR    000014        IN.DAT=  000400        L$CHTB   000030        L.LLA    000002
C.CNT    000010           E$NBS    000020        IN.ILS=  000001        L$CLEN=  000044        L.LNG    000150
C.DELF   000016           E$NCR    000034        IO.WVB=  ******  GX    L$CNT =  ******  GX    L.LNGX=  000150  G
C.DELW   000017           E$NCS    000036        LA.ACK=  100000        L$CTEN   000032        L.LNO    000026
C.FLG    000006           E$NIC    000044        LA.CRS=  020000        L$CTIM   000040        L.LPT    000065
C.INAC   000014           E$NLEN   000050        LA.MSK=  170000        L$DTEA   000020        L.LSA    000030
C.INCT   000012           E$NLLA   000012        LA.NAK=  110000        L$DTEL   000017        L.LSFD   000046
C.IPL    000020           E$NLNK   000000        LA.NMS=  010000        L$GLEN   000134        L.LSFI   000044
C.LKS    000010           E$NML    000040        LA.RES=  040000        L$LEN    000122        L.LTT    000062
C.LST    000021           E$NMR    000024        LA.WND=  004000        L$LENX=  000124  G     L.MASQ   000070
C.NAM    000004           E$NMS    000030        LF.DRD=  000004        L$LLCH   000016        L.MAST   000073
C.NDC    000006           E$NNOD   000002        LF.FRC=  000001        L$LNK    000000        L.MASZ   000072
C.OUTT   000013           E$NRT    000042        LF.HFO=  000010        L$MCHN   000036        L.NIN    000020
C.PRM =  ******  GX       E$NRTP   000005        LF.HMF=  000040        L$NAM =  ******  GX    L.NXN    000016
C.RETF   000015           E$NSEG   000010        LF.HSF=  000020        L$NETW   000114        L.NXTH   000010
C.SEG    000004           E$NTIM   000046        LF.IRD=  000002        L$NIRE   000072        L.OPD    000103
C.STS =  ******  GX       E$NUSE   000004        LF.MFL=  ******  GX    L$NLRE   000104        L.OPDL   000102
C.UNT    000011           E$STRT   000006        LF.MMF=  000200        L$NMAC   000076        L.PCTL   000144
D$ACL    000027           FF.ADD=  000020        LF.MSF=  000100        L$NMAS   000074        L.PDVC   000126
D$CUGL   000023           FF.CIR=  000040        LF.MTP=  ******  GX    L$NMST   000003        L.PDVD   000127
D$CUGN   000022           FF.CON=  000001        LN$OFF=  000001        L$NNRE   000106        L.REM    000006
D$DATL   000030           FF.FIL=  000002        LN$ON =  000000        L$NRBY   000046        L.RFC    000050
D$DSL    000022           FF.HST=  040000        LN$SHU=  000002        L$NRCA   000066        L.RLA    000004
D$DST    000010           FF.LIN=  000010        LP$EIP=  002000        L$NRFS   000100        L.RNO    000022
```

```
    629 000004                          D$ANUM::.BLKW   5               ; REMOTE DTE NUMBER (1.-15. DECIMAL DIGITS)
    630 000016                          D$ASLO::.BLKW   1               ; SUBADDRESS LOW RANGE
    631 000020                          D$ASHI::.BLKW   1               ; SUBADDRESS HI RANGE
    632 000022                          D$ACUG::.BLKW   2               ; USER GROUP NAME
    633 000026                          D$ADLN::.BLKW   1               ; LENGTH OF BCD DTE ADDRESS IN BYTES
    634                 000030          D$ALEN==.
    635 002054                                  .PSECT
    636
    637                                 ;****
    638                                 ;
    639                                 ; DEFINE THE DESTINATION CALL PARAMETER BLOCK
    640                                 ;
    641 000030                                  .ASECT
    642                 000000                  .=0
    643 000000                                  .BLKW   1               ; C.LNK
    644 000002                                  .BLKW   1               ; C.STS
    645 000004                          D$CMSK::.BLKW   11.             ; CALL MASK (1.-32. HEX DIGITS)
    646 000032                          D$CVAL::.BLKW   11.             ; CALL VALUE (1.-32. HEX DIGITS)
    647 000060                          D$CMLN::.BLKB   1               ; LENGTH OF CALL MASK
    648 000061                          D$CVLN::.BLKB   1               ; LENGTH OF CALL VALUE
    649                 000062          D$CLEN==.
    650 002054                                  .PSECT
    651
    652                                 ;****
    653                                 ;
    654                                 ; DEFINE THE RDT PARAMETER BLOCK
    655                                 ;
    656 000062                                  .ASECT
    657                 000000                  .=0
    658 000000                                  .BLKW   1               ; C.LNK
    659 000002                                  .BLKW   1               ; C.STS
    660 000004                          R$DNAM::.BLKW   2.              ; RAD50 DTE NAME
    661 000010                          R$DADR::.BLKW   5.              ; RAD50 DTE ADDRESS (1.-15. DECIMAL DIGITS)
    662                 000022          R$DLEN==.
    663 002054                                  .PSECT
    664                                         .NLIST  BEX
    665
```

```
                                                                  D 2
CFEIMP - SYMBOL DEFINITIONS AND MACRO V05.03b Saturday 29-Jun-85 04:55  Page 13-1
Symbol table

A.LULU= 000002                   CS.EVT= 000017 G                 C$UNAM  000004 G                 ERRLIC  002445RG                 F.CNTG= 000034
A.LUNA= 000004                   CS.FEA= 000015 G                 C.CMLD= 000002                   ERRNOD  002554RG                 F.DFNB= 000046
A.LUNU= 000006                   CS.FIL= 000045 G                 C.DEVD= 000006                   ERROUT  002303RG                 F.DSPT= 000044
BUFSIZ= 000764                   CS.FLG= 010000 G                 C.DIRD= 000012                   ERRPNA  002616RG                 F.DVNM= 000134
CH$HI = 000400 G                 CS.INP= 000001                   C.DSDS= 000006                   ERRPRC  002400RG                 F.EFBK= 000010
CL.AST= 004000                   CS.LLC= 000003 G                 C.FILD= 000016                   ERRSYN  002054RG                 F.EFN = 000050
CL.ATT= 000200                   CS.LOG= 000020 G                 C.LNK   000000 G                 EX$ERR= 000002                   F.EOBB= 000032
CL.CAS= 020000                   CS.MOD= 100000 G                 C.MKW1= 000024                   EX$SEV= 000004                   F.ERR = 000052
CL.CMT= 000001 G                 CS.MOR= 000020                   C.MKW2= 000026                   EX$SUC= 000001                   F.EXT = 000072
CL.EIF= 000040 G                 CS.MSK= 000400 G                 C.PRM   000004 G                 EX$WAR= 000000                   F.FACC= 000043
CL.EXN= 040000                   CS.NET= 000400 G                 C.SIZE= 000054                   E2.BEA= 000002 G                 F.FFBY= 000014
CL.EXT= 000010                   CS.NMF= 000001                   C.STAT= 000001                   E2.BRT= 000001 G                 F.FLG = 000074
CL.IND= 000002                   CS.NOD= 000012 G                 C.STS   000002 G                 E2.INA= 000010 G                 F.FNAM= 000110
CL.LCC= 000020 G                 CS.NUM= 000400 G                 C.SWAD= 000022                   E2.INC= 000020 G                 F.FNB = 000102
CL.NPR= 000400                   CS.OBJ= 000014 G                 C.TYPR= 000000                   E2.NSP= 000174 G                 F.FTYP= 000116
CL.OPN= 000100 G                 CS.OUT= 000002                   DA$CUG= 004000 G                 E2.OUT= 000040 G                 F.FVER= 000120
CL.RNE= 000004 G                 CS.PAR= 000011 G                 DEFFLG= 000233                   E2.RET= 000100 G                 F.HIBK= 000004
CL.XIT= 100000 G                 CS.PSN= 000022 G                 DFLTNB  000426R              003 E2.ROU= 000003 G                 F.LUN = 000042
CM.DEF= 000001 G                 CS.PVC= 000030 G                 DFTLEN= 000007                   E2.SEG= 000004 G                 F.MBCT= 000054
CM.EXI= 000004 G                 CS.RDT= 000035 G                 D$ACUG  000022 G                 FD.CR = ****** GX                F.MBC1= 000055
CM.HLP= 000003 G                 CS.REM= 000013 G                 D$ADLN  000026 G                 FL.ADD= 002000 G                 F.MBFG= 000056
CM.KIL= 000005 G                 CS.ROU= 000016 G                 D$ALEN= 000030 G                 FL.CRE= 001000 G                 F.NRBD= 000024 G
CM.LIS= 000002 G                 CS.SAC= 000041 G                 D$ANUM  000004 G                 FL.DDM= 000001 G                 F.NREC= 000030
CM.PUR= 000006 G                 CS.SAD= 000042 G                 D$ASHI  000020 G                 FL.DLC= 000002 G                 F.OVBS= 000030
CO.CIR= 000010 G                 CS.SER= 000044 G                 D$ASLO  000016 G                 FL.DLM= 000200 G                 F.RACC= 000016
CO.EXE= 000003 G                 CS.SHI= 001000 G                 D$CLEN= 000062 G                 FL.KMX= 000020 G                 F.RATT= 000001
CO.KLG= 000007 G                 CS.SLO= 002000 G                 D$CMLN  000060 G                 FL.LLC= 000004 G                 F.RCNM= 000034
CO.KNC= 000011 G                 CS.SLT= 000002 G                 D$CMSK  000004 G                 FL.LMC= 000100 G                 F.RCTL= 000017
CO.KNL= 000006 G                 CS.SNA= 000040 G                 D$CVAL  000032 G                 FL.MFL= 000010 G                 F.RSIZ= 000002
CO.KNN= 000001 G                 CS.SPU= 000043 G                 D$CVLN  000061 G                 FL.MTP= 000400 G                 F.RTYP= 000000
CO.KNO= 000005 G                 CS.STA= 000010 G                 D$DFLG  000006 G                 FL.MUX= 000040 G                 F.SEQN= 000100
CO.KNP= 000002 G                 CS.SVC= 000021 G                 D$DLEN= 000014 G                 FL.NAM= 004000 G                 F.SPDV= 000072
CO.LIN= 000006 G                 CS.THI= 000400 G                 D$DNAM  000004 G                 FL.PVC= 004000 G                 F.SPUN= 000074
CO.LOG= 000007 G                 CS.TLO= 001000 G                 D$DNCT  000013 G                 FL.SLV= 001000 G                 F.STBK= 000036
CO.MAC= 000014 G                 CS.TOP= 000400 G                 D$DNLT  000012 G                 FL.SNA= 040000 G                 F.UNIT= 000136
CO.MSP= 000016 G                 CS.TRB= 020000 G                 D$DPRI  000010 G                 FL.TIM= 000200 G                 F.URBD= 000020 G
CO.MXP= 000012 G                 CS.TSK= 000400 G                 D$SLEN= 000020 G                 FL.TRB= 020000 G                 F.VBN = 000064
CO.MXS= 000013 G                 CS.UNI= 040000 G                 D$SNAM  000006 G                 FL.TRI= 010000 G                 F.VBSZ= 000060
CO.MX9= 000015 G                 CS.UNT= 000007 G                 D$SOBJ  000013 G                 FL.X25= 002000 G                 G.MCRB= 000002 G
CO.NOD= 000001 G                 CS.VAL= 001000 G                 D$SPRI  000012 G                 FO.UPD= ****** GX                IN$CHA= 000004 G
CO.OBJ= 000005 G                 CS.WLD= 000010                   D$STSK  000014 G                 FO.WRT= ****** GX                IN$COU= 000006 G
CO.PRO= 000002 G                 CS.X2P= 000024 G                 D$STYP  000004 G                 FX.DIR= 000004                   IN$EVE= 000010 G
CO.SYS= 000004 G                 CS.X29= 000025 G                 D$TADR  000004 G                 FX.DLN= 000003                   IN$STA= 000002 G
CS.BUF= 000001 G                 CS.X3P= 000023 G                 D$TCT   000032 G                 FX.DMX= 000002                   IN$SUM= 000000 G
CS.CHN= 000027 G                 CU$BIL= 000001 G                 D$THSH  000030 G                 FX.LEN= 000000                   IO.WVB= ****** GX
CS.CNT= 000006 G                 C$HHI   000006 G                 D$TLEN= 000040 G                 F.ACTL= 000076                   LF.BRO= 000100 G
CS.CUG= 000031 G                 C$HLEN= 000010 G                 D$TLIN  000016 G                 F.ALOC= 000040                   LF.ENA= 000001 G
CS.DDM= 000005 G                 C$HLO   000004 G                 D$TNET  000034 G                 F.BBFS= 000062                   LF.MDC= 000002 G
CS.DEC= 000046 G                 C$NCNT  000004 G                 D$TSTA  000026 G                 F.BDB = 000070                   LF.MFL= 000010 G
CS.DIF= 000002                   C$NCSR  000010 G                 EF    = 000001                   F.BGBC= 000057                   LF.MTP= 000004 G
CS.DLC= 000004 G                 C$NLEN= 000016 G                 ERRALL  002336RG                 F.BKDN= 000026                   LF.SER= 000200 G
CS.DSA= 000033 G                 C$NPRI  000012 G                 ERRCIS  002507RG                 F.BKDS= 000020                   LF.TIM= 000020 G
CS.DSC= 000034 G                 C$NURM  000014 G                 ERRCON  002101RG                 F.BKEF= 000050                   LF.X2P= 000040 G
CS.DST= 000032 G                 C$NVEC  000006 G                 ERRFIL  002245RG                 F.BKP1= 000051                   LUNX  = 000006
CS.DTE= 000026 G                 C$UCUG  000010 G                 ERRGCL  002163RG                 F.BKST= 000024                   LUN1  = 000001
CS.DVF= 000004                   C$UFLG  000012 G                 ERRILC  002133RG                 F.BKVB= 000064                   LUN2  = 000002
CS.EQU= 000040                   C$ULEN= 000013 G                 ERRILN  002216RG                 F.CHR = 000075                   LUN3  = 000003
```

```
                                                                        D  3
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 10
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE              REFERENCES

UN$SCS  = 000400  G        #12-1196
U$NCSR    000012  G        #9-408
U$NCST    000014  G        #9-409
U$NCW0    000006  G        #9-406
U$NCW1    000010  G        #9-407
U$NDPR    000016  G        #9-410
U$NLEN  = 000026  G        #9-414
U$NPCH    000024  G        #9-413
U$NUNT    000004  G        #9-405
WC.CIR  = 000017  G        #11-753
WC.CNT  = 000002  G        #11-749
WC.EVE  = 000001  G        #11-757
WC.LIN  = 000017  G        #11-752
WC.LOG  = 000002  G        #11-758
WC.NAM  = 000001  G        #11-748
WC.NOD  = 000001  G        #11-756
WC.OBJ  = 000001  G        #11-755
WC.PRO  = 000001  G        #11-754
WC.TRI  = 000010  G        #11-751
WC.UNT  = 000004  G        #11-750
W.TSEF  = 000002            6-95
XA.DST  = 000001  G        #11-998
XA.KND  = 000002  G        #11-999      11-1000
XA.NUM  = 000004  G        #11-1000
XP.BIL  = 000004  G        #11-975
XP.CHN  = 000001  G        #11-961
XP.CLT  = 000002  G        #11-941      11-953
XP.COU  = 000002  G        #11-962
XP.CTM  = 000001  G        #11-940      11-953
XP.CUG  = 040000  G        #11-970
XP.DFB  = 000004  G        #11-942      11-953
XP.DFW  = 000010  G        #11-943      11-953
XP.DTA  = 000001  G        #11-973
XP.DTE  = 010000  G        #11-958
XP.KND  = 020000  G        #11-959
XP.KNG  = 100000  G        #11-971
XP.LIN  = 000004  G        #11-963
XP.LV3  = 003777  G        #11-953      11-954     #11-954
XP.MCI  = 000010  G        #11-964
XP.MRE  = 000100  G        #11-946      11-953
XP.MXB  = 000020  G        #11-944      11-953
XP.MXC  = 000040  G        #11-945      11-953
XP.MXR  = 000200  G        #11-947      11-954
XP.MXW  = 000400  G        #11-948      11-954
XP.NNA  = 004000  G        #11-951
XP.NUM  = 000002  G        #11-974
XP.RET  = 001000  G        #11-949      11-954
XP.RST  = 002000  G        #11-950      11-954
XP.STA  = 000020  G        #11-965      11-979
XP.ST1  = 000040  G        #11-966      11-980
XS.CLM  = 000020  G        #11-987      11-1009
XS.CLV  = 000040  G        #11-988      11-1010

                                                                        E  3
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 11
```

D 4

```
CFEINI - ONCE ONLY INITIALIZATI MACRO V05.03b Saturday 29-Jun-85 04:56  Page 5-1
Symbol table

CL.CMT= ****** GX     PRMPT    000000R        $CLIOS= ****** GX     $IOPEN= ****** GX     $XXINI= ****** GX
ERRFIL= ****** GX     $BUFF = ****** GX       $CLPMT= ****** GX     $LOVBN= ****** GX     .CLOSE= ****** G
ERRGCL= ****** GX     $BYTE = ****** GX       $EXSTS= ****** GX     $OFDB = ****** GX     .DELET= ****** G
EX$SUC= ****** GX     $CFERR= ****** GX       $GCLEP= ****** GX     $OPEN = ****** GX     .FINIT= ****** G
IE.EOF= ****** GX     $CFINI   000024RG       $HIVBN= ****** GX     $TMSTB= ****** GX      MARK = ****** GX
PAR$$$= 000027        $CLFLG= ****** GX       $IFDB = ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000254    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8840 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.19
SY:CFEINI.V2,[132,134]CFEINI/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEINI
```

E 4

```
     58                                         ;****
     59                                         ; LIBRARY MACRO CALLS
     60                                         ;****
     61
     62                                                 .MCALL  CSI$1,CSI$2,FDOP$R,OPEN$W,PUT$,CSI$
     63                                                 .MCALL  GET$,OPEN$R,FDOF$L,NBOF$L,QIOW$,DIR$
     64
     65                                         ;****
     66                                         ; LOCAL DATA
     67                                         ;****
     68
     69 000000                                          CSI$                            ; DEFINE CSI OFFSETS
     70 000000                                          FDOF$L                          ; DEFINE FDB OFFSETS
     71 000000                                          NBOF$L                          ; DEFINE NAME BLOCK OFFSETS
     72
     73 000000                                  RATDPB: QIOW$   IO.RAT,0,1,,IOSB,,<$IFDB+F.FNB+N.FID,$FIATR>
     74 000030                                  IOSB:   .BLKW   2
     75
     76                 000122                          BUFSZ=82.                       ; OUTPUT BUFFER SIZE
     77 000034                                  $BUFR:: .BLKB   BUFSZ                   ; OUTPUT FILE BUFFER
     78 000156                                  $BUFI:: .BLKB   BUFSZ                   ; INPUT FILE BUFFER
     79 000300                                  CSBLK:  .BLKB   C.SIZE                  ; CSI CONTROL BLOCK
     80                                                 .EVEN
```

```
    129                                        .SBTTL  LOCAL SYMBOL DEFINITIONS
    130                                ;****
    131                                ; LOCAL SYMBOL DEFINITIONS
    132                                ;****
    133
    134          000013                OBJLEN = 13                             ; LENGTH OF OBJECT TEMPLATE
    135                                ;
    136                                ; DEFINE EVENT TEMPLATE BLOCK
    137                                ;
    138 000040                                 .ASECT
    139          000000                        .=0
    140 000000                                 .BLKW   1                       ; LINK WORD
    141 000002                                 .BLKW   1                       ; STATUS
    142 000004                         C.CLS:  .BLKW   1                       ; EVENT CLASS
    143 000006                         C.EVT:  .BLKW   4                       ; EVENT MASKS
    144 000016                         C.SNK:  .BLKW   1                       ; SINK TYPE
    145 000020                         EVTLEN:
    146 000214                                 .PSECT
    147
```

```
**FILE**ID**CFEPRS





  CCCCCCCC    FFFFFFFFFF  EEEEEEEEEE  PPPPPPPP    RRRRRRRR      SSSSSSSS
  CCCCCCCC    FFFFFFFFFF  EEEEEEEEEE  PPPPPPPP    RRRRRRRR      SSSSSSSS
CC            FF          EE          PP      PP  RR      RR  SS
CC            FF          EE          PP      PP  RR      RR  SS
CC            FF          EE          PP      PP  RR      RR  SS
CC            FF          EE          PP      PP  RR      RR  SS
CC            FFFFFFFF    EEEEEEEE    PPPPPPPP    RRRRRRRR      SSSSSS
CC            FFFFFFFF    EEEEEEEE    PPPPPPPP    RRRRRRRR      SSSSSS
CC            FF          EE          PP          RR  RR              SS
CC            FF          EE          PP          RR  RR              SS
CC            FF          EE          PP          RR    RR            SS         ....
CC            FF          EE          PP          RR    RR            SS         ....
  CCCCCCCC    FF          EEEEEEEEEE  PP          RR      RR  SSSSSSSS           ....
  CCCCCCCC    FF          EEEEEEEEEE  PP          RR      RR  SSSSSSSS           ....




LL              SSSSSSSS  TTTTTTTTTT
LL              SSSSSSSS  TTTTTTTTTT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL              SSSSSS        TT
LL              SSSSSS        TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LLLLLLLLLL    SSSSSSSS        TT
LLLLLLLLLL    SSSSSSSS        TT
```

```
000000                                   TRAN$  $LAMDA
000000                                  STATE$  PNSPA2
000000                                   TRAN$  &K$PAS, $EXIT
                                ;
                                ; Purge Secondary [loader]
                                ;
000000                                  STATE$  PNSLO
000000                                   TRAN$  &K$SEC
000000                                  STATE$
000000                                   TRAN$  &K$LOAD,$EXIT
000000                                   TRAN$  $LAMDA, $EXIT

                                ;
                                ; Purge Tertiary [loader]
                                ;
000000                                  STATE$  PNTLO
000000                                   TRAN$  &K$TER
000000                                  STATE$
000000                                   TRAN$  &K$LOAD,$EXIT
000000                                   TRAN$  $LAMDA, $EXIT

                                ;
                                ; Purge Load file ( Load [file] or [Load] file)
                                ;
000000                                  STATE$  PNLOD
000000                                   TRAN$  &K$LOAD,PNLOD2
000000                                   TRAN$  $LAMDA, PNLOD3

000000                                  STATE$  PNLOD2
000000                                   TRAN$  &K$FIL, $EXIT
000000                                   TRAN$  $LAMDA, $EXIT

000000                                  STATE$  PNLOD3
000000                                   TRAN$  &K$FIL, $EXIT

                                ;
                                ; Purge Diagnostics [file]
                                ;
000000                                  STATE$  PNDIA
000000                                   TRAN$  &K$DIAG

000000                                  STATE$
000000                                   TRAN$  &K$FIL, $EXIT
000000                                   TRAN$  $LAMDA, $EXIT

                                ;
                                ; Purge Dump file
                                ;
000000                                  STATE$  PNDMP
000000                                   TRAN$  &K$DMP

000000                                  STATE$  &P$DMPO
000000                                   TRAN$  &K$FIL, $EXIT

                                ;
                                ; LOGGING OPTIONS
                                ;
```

```
   1208 000000                                   TRAN$  $DNUMB, $EXIT,  .CIRPR
   1209
   1210                                ;
   1211                                ; CIRCUIT SERVICE [ENABLED/DISABLED]
   1212                                ;
   1213 000000                                  STATE$  CISER
   1214 000000                                   TRAN$  &K$SER
   1215 000000                                  STATE$  ,&P$SER
   1216 000000                                   TRAN$  &K$ENA, $EXIT,,         SW.SEN, $SWICH
   1217 000000                                   TRAN$  &K$DIS, $EXIT,,         SW.SDI, $SWICH
   1218 000000                                   TRAN$  $LAMDA, $FAIL,  $CLERR, ERRIPV, $ERROR
   1219
   1220                                ;
   1221                                ; LOOK FOR CIRCUIT STATE
   1222                                ;
   1223 000000                                  STATE$  CSTATE
   1224 000000                                   TRAN$  &K$STA
   1225
   1226 000000                                  STATE$  ,&P$CIST
   1227 000000                                   TRAN$  &K$ON,   $EXIT,, SW.LOA!SW.ENA,  $SWICH
   1228 000000                                   TRAN$  &K$OFF,  $EXIT,, SW.LOA,         $SWICH
   1229                                ;
   1230                                ; LOOK FOR CIRCUIT COST
   1231                                ;
   1232 000000                                  STATE$  COST
   1233 000000                                   TRAN$  &K$COS
   1234
   1235 000000                                  STATE$  ,&P$COS
   1236 000000                                   TRAN$  $DNUMB,$EXIT,,.CICOS
   1237                                ;
   1238                                ; LOOK FOR TRIBUTARY ADDRESS
   1239                                ;
   1240 000000                                  STATE$  TRIB
   1241 000000                                   TRAN$  &K$TRIB
   1242
   1243 000000                                  STATE$  ,&P$TRIB
   1244 000000                                   TRAN$  $DNUMB,$EXIT,,.CITRI
   1245
   1246                                ;
   1247                                ; MULTI-POINT ACTIVE
   1248                                ;
   1249 000000                                  STATE$  APR
   1250 000000                                   TRAN$  &K$MUL
   1251
   1252 000000                                  STATE$  ,&P$MUL
   1253 000000                                   TRAN$  &K$ACT
   1254
   1255 000000                                  STATE$  ,&P$ACT
   1256 000000                                   TRAN$  $DNUMB,$EXIT,,.CIACT
   1257                                ;
   1258                                ; CHANNEL NUMBER
   1259                                ;
   1260 000000                                  STATE$  CHN
   1261 000000                                   TRAN$  &K$CHN
   1262 000000                                  STATE$  ,&P$CHN
   1263 000000                                   TRAN$  $DNUMB,$EXIT,,.CICHN
   1264                                ;
```

```
                                        .SBTTL  DEFINE OBJECT COMMAND
                              ;
                              ; DEFINE OBJECT COMMAND
                              ;
000000                                  STATE$  DEFOBJ
000000                                   TRAN$  !OBJID,,,       CO.OBJ, $COMP

000000                                  STATE$  ,&P$DEOB
000000                                   TRAN$  !OBJOPT

000000                                  STATE$  OBJ0
000000                                   TRAN$  !OBJOPT,OBJ0
000000                                   TRAN$  $LAMDA, $EXIT
                              ;
                              ; OBJECT OPTIONS
                              ;
000000                                  STATE$  OBJOPT
000000                                   TRAN$  !ONAME, $EXIT,, OO.NAM, $OPRND
000000                                   TRAN$  !COPY,  $EXIT,, OO.COP, $OPRND
000000                                   TRAN$  !USER,  $EXIT,, OO.USR, $OPRND
000000                                   TRAN$  !OVFY,  $EXIT,, OO.VFY, $OPRND
                              ;
                              ; OBJECT NAME
                              ;
000000                                  STATE$  ONAME
000000                                   TRAN$  &K$NAM

000000                                  STATE$  ,&P$ONAM
000000                                   TRAN$  $RAD50, $EXIT,  .OBNAM
                              ;
                              ; COPIES
                              ;
000000                                  STATE$  COPY
000000                                   TRAN$  &K$COP

000000                                  STATE$  ,&P$COP
000000                                   TRAN$  &K$SIN, $EXIT
000000                                   TRAN$  $DNUMB, $EXIT,  .OBCOP, SW.SMC, $SWICH
                              ;
                              ; USER (DEFAULT/LOGIN)
                              ;
000000                                  STATE$  USER
000000                                   TRAN$  &K$USE

000000                                  STATE$  ,&P$USE
000000                                   TRAN$  &K$DFA, $EXIT
000000                                   TRAN$  &K$LOGI,$EXIT,, SW.RLU, $SWICH
                              ;
                              ; VERIFICATION
                              ;
000000                                  STATE$  OVFY
000000                                   TRAN$  &K$VER

000000                                  STATE$  ,&P$OVER
000000                                   TRAN$  &K$ON,  $EXIT
000000                                   TRAN$  &K$OFF, $EXIT,, 2,      $VLVL
000000                                   TRAN$  &K$INS, $EXIT,, 1,      $VLVL
```

```
  2541 000000                                          STATE$
  2542 000000                                           TRAN$  &K$SRV, BLAUT
  2543 000000                                           TRAN$  $LAMDA,,        .BL.AU3
  2544                                         ;
  2545                                         ; SUBEXPRESSION TO MATCH MODULE X25-ACCESS
  2546                                         ;
  2547 000000                                          STATE$  MODXAC
  2548 000000                                           TRAN$  !MX25
  2549 000000                                          STATE$
  2550 000000                                           TRAN$  &K$ACC, BLAUT
  2551 000000                                           TRAN$  $LAMDA,,        .BL.AU3
  2552                                         ;
  2553                                         ; SUBEXPRESSION TO MATCH MODULE X29-SERVER
  2554                                         ;
  2555 000000                                          STATE$  MODX29
  2556 000000                                           TRAN$  !MX29
  2557 000000                                          STATE$
  2558 000000                                           TRAN$  &K$SRV, BLAUT
  2559 000000                                           TRAN$  $LAMDA,,        .BL.AU3
  2560                                         ;
  2561                                         ; SUBEXPRESSION TO TURN ON AUTOMATIC BLANK PARSING
  2562                                         ;
  2563 000000                                          STATE$  BLAUT
  2564 000000                                           TRAN$  $LAMDA, $EXIT,  $BL.A3T
  2565                                         ;
  2566                                         ; SUBEXPRESSION TO MATCH "[MODULE] X25-"
  2567                                         ;
  2568 000000                                          STATE$  MX25
  2569 000000                                           TRAN$  &K$MOD, MX25A
  2570 000000                                           TRAN$  $LAMDA, MX25B
  2571 000000                                          STATE$  MX25A,  &P$MOD
  2572 000000                                           TRAN$  $EOS,   MX25A
  2573 000000                                           TRAN$  $LAMDA
  2574 000000                                          STATE$  MX25B
  2575 000000                                           TRAN$  &K$X25
  2576 000000                                          STATE$
  2577 000000                                           TRAN$  $LAMDA,,        $BL.MN1
  2578 000000                                          STATE$
  2579 000000                                           TRAN$  <'->,   $EXIT
  2580                                         ;
  2581                                         ; SUBEXPRESSION TO MATCH "[MODULE] X29-"
  2582                                         ;
  2583 000000                                          STATE$  MX29
  2584 000000                                           TRAN$  &K$MOD, MX29A
  2585 000000                                           TRAN$  $LAMDA, MX29B
  2586 000000                                          STATE$  MX29A,  &P$MOD
  2587 000000                                           TRAN$  $EOS,   MX29A
  2588 000000                                           TRAN$  $LAMDA
  2589 000000                                          STATE$  MX29B
  2590 000000                                           TRAN$  &K$X29
  2591 000000                                          STATE$
  2592 000000                                           TRAN$  $LAMDA,,        $BL.MN1
  2593 000000                                          STATE$
  2594 000000                                           TRAN$  <'->,   $EXIT
  2595                                         ;
  2596                                         ; MAXIMUM WINDOW
  2597
```

```
**FILE**ID**CFEROT





  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   RRRRRRRR       000000      TTTTTTTTTT
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   RRRRRRRR       000000      TTTTTTTTTT
CC           FF           EE           RR      RR   00      00        TT
CC           FF           EE           RR      RR   00      00        TT
CC           FF           EE           RR      RR   00      00        TT
CC           FF           EE           RR      RR   00      00        TT
CC           FFFFFFFF     EEEEEEEE     RRRRRRRR     00      00        TT
CC           FFFFFFFF     EEEEEEEE     RRRRRRRR     00      00        TT
CC           FF           EE           RR  RR       00      00        TT
CC           FF           EE           RR  RR       00      00        TT
CC           FF           EE           RR    RR     00      00        TT            ....
CC           FF           EE           RR    RR     00      00        TT            ....
  CCCCCCCC   FF           EEEEEEEEEE   RR      RR     000000          TT            ....
  CCCCCCCC   FF           EEEEEEEEEE   RR      RR     000000          TT            ....




LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

E 12

CFEROT - ROOT SEGMENT FOR CFF   MACRO V05.03b Saturday 29-Jun-85 05:13

```

                                                 .SBTTL  $RQBF - REQUEST CORE BLOCK WITH TASK EXTENSION
                                         ;+
                                         ; **-$RQBF-REQUEST CORE BLOCK WITH TASK EXTENSION
                                         ;
                                         ; THIS ROUTINE IS CALLED TO REQUEST A CORE BLOCK FROM THE FREE LIST.
                                         ; THE SELECTION IS MADE EITHER FIRST FIT OR BEST FIT ACCORDING TO THE
                                         ; THE INPUT PARAMETERS. IF THERE IS NOT ENOUGH SPACE IN THE BUFFER POOL,
                                         ; AN ATTEMPT IS MADE TO EXTEND THE TASK TO INCREASE THE AVAILABLE BUFFERS.
                                         ;
                                         ;
                                         ; INPUTS:
                                         ;
                                         ;        R0=ADDRESS OF FREE BLOCK LIST HEAD.
                                         ;        R1=SIZE OF BLOCK REQUESTED IN BYTES.
                                         ;                IF R1 IS GE 0, THEN BEST FIT IS USED.
                                         ;                IF R1 IS LT 0, THEN FIRST FIT IS USED.
                                         ;
                                         ;        IN EITHER CASE THE MAGNITUDE OF R1 IS THE SIZE OF THE BLOCK.
                                         ;
                                         ;        $XXBUF  = ADDRESS OF BUFFER
                                         ;        $XXLEN  = LENGTH OF BUFFER POOL
                                         ;
                                         ; OUTPUTS:
                                         ;
                                         ;        C=1 IF REQUEST CANNOT BE FULFILLED.
                                         ;        C=0 IF REQUEST IS SATISFIED.
                                         ;                R0=ADDRESS OF REQUESTED CORE BLOCK.
                                         ;
                                         ;                $XXBUF, $XXLEN ARE UPDATED,
                                         ;
                                         ;-

 000000                                  $RQBF::
 000000                                  10$:    SAVRG   <R0,R1,R2>       ; SAVE PARAMETERS IN CASE OF FAILURE
 000006  004767  000174                          CALL    .RQCB            ; TRY TO ALLOCATE OUT OF BUFFER POOL
 000012  103030                                  BCC     20$              ; IF CC, SUCCESS
 000014                                          EXTK$S  #.EXINC          ; ELSE TRY TO EXTEND TASK
 000030  103414                                  BCS     15$              ; IF CS, DIRECTIVE FAILED
 000032  016600  000004                          MOV     4(SP),R0         ; ELSE POINT TO FREE LISTHEAD.
 000036  016702  000000G                         MOV     $XXBUF,R2        ; CALCULATE ADDRESS OF
 000042  066702  000000G                         ADD     $XXLEN,R2        ;  EXTENSION
 000046  012701  002000                          MOV     #.EXINC*64.,R1   ; GET SIZE OF EXTENSION
 000052  060167  000000G                         ADD     R1,$XXLEN        ; EXTEND BUFFER POOL SIZE
 000056  004767  000020                          CALL    $RLBF            ; LET CORAL PUT IT ON LIST
 000062                                  15$:    RESRG   <R2,R1,R0>       ; RESTORE SAVED REGISTERS
 000070  103343                                  BCC     10$              ; IF CC, TRY AND ALLOCATE AGAIN
 000072  000207                                  RETURN                   ; ELSE RETURN FAILURE

 000074  022626                          20$:    CMP     (SP)+,(SP)+      ; CLEAN UP STACK ...
 000076  005726                                  TST     (SP)+            ; ... RETURN CLEAR CARRY
 000100  000207                                  RETURN
```

```
                                        .SBTTL  $TALOC - ALLOCATE TEMPLATES
                                ;+
                                ; *** - $TALOC - ALLOCATE TEMPLATES OR EXIT ON ERROR
                                ;
                                ; INPUTS:
                                ;       R1 = LENGTH OF ALLOCATION REQUEST
                                ;
                                ; OUTPUTS:
                                ;       R0 = ADDRESS OF ALLOCATED BUFFER
                                ;
                                ;-

000146                          $TALOC::
000146  012700  000000G                 MOV     #$XXAVL,R0              ; GET ADDRESS OF END OF TASK LISTHEAD
000152  004767  000012                  CALL    $T1ALC                  ; TRY TO ALLOCATE BLOCK
000156  012701  000000G                 MOV     #$TMLST,R1              ; GET TEMPLATE LISTHEAD
000162  004767  000000G                 CALL    $QINS                   ; INSERT TEMPLATE INTO LIST
000166  000207                          RETURN
```

```
                                        .SBTTL  $MVASC - MOVE ASCII STRING
                                ;+
                                ; *** - $MVASC - MOVE ASCII STRING
                                ;
                                ; SUBROUTINE TO MOVE ASCII STRINGS
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = OUTPUT BUFFER POINTER
                                ;       R1 = INPUT BUFFER POINTER
                                ;       R2 = NUMBER OF BYTES TO MOVE
                                ;
                                ; OUTPUTS:
                                ;       R0 = NEXT LOCATION IN OUTPUT BUFFER
                                ;       R1 = NEXT LOCATION IN INPUT BUFFER
                                ;       R2 = DESTROYED
                                ;
                                ;-

000164                          $MVASC::
000164  005702                          TST     R2              ; EMPTY STRING ?
000166  003403                          BLE     20$             ; IF LE, YES - LEAVE
000170  112120                  10$:    MOVB    (R1)+,(R0)+     ; MOVE THE NEXT BYTE
000172  005302                          DEC     R2              ; ALL DONE ?
000174  003375                          BGT     10$             ; IF GT, NO - LOOP
000176  000207                  20$:    RETURN
```

```
Symbol table

L.RTQ   000060        O.TYP   000002        S$TIM   000014        US$CNF= 000002        X$PRT   000021
L.RTYD  000055        PRINT   000014R       S$TMR   000016        US$DIS= 000006        X$PS    000010
L.RTYI  000057        Q.IOFN= ****** GX     S$WND   000006        US$DON= 000000        X$RPR   000013
L.SEC   000064        Q.IOPL= ****** GX     S.ADD   000004        US$DSC= 000004        X$RPS   000012
L.SEGZ  000076        RF.LOO= 100000        S.CIR   000016        US$EAC= 000012        X$RTRY  000002
L.STA   000000        R$DAL   000013        S.DEV   000020        US$WDS= 000010        X$RXI   000104
L.TC    000042        R$DTE   000014        S.DIA   000046        WC.CNT= ****** GX     X$RXQ   000072
L.TIC   000043        R$LEN   000024        S.DPA   000032        WC.NAM= ****** GX     X$SS    000004
L.TIPD  000013        R$LENX= 000024 G      S.DPC   000036        WC.TRI= ****** GX     X$ST    000005
L.TIPI  000012        R$LNK   000000        S.DUM   000050        WC.UNT= ****** GX     X$TCLZ  000030
L.TMRD  000054        R$NAM   000002        S.FLAG  000002        XA$NCO= 000020        X$TIMC  000001
L.TMRI  000056        R$OWN   000010        S.HAD   000010        XA$RR = 000200        X$TIMR  000000
L.TYP   000001        R.ADD   000010        S.HST   000006        XA$UCL= 000040        X$TXI   000102
L.ULA   000125        R.FLAG  000012        S.LENG  000052        XA$URE= 000100        X$TXQ   000066
L.USA   000024        R.LEN   000014        S.LENX= 000054 G      XF$NCO= 000020        X$TYP   000015
L.USTA  000036        R.LENX= 000014 G      S.LNK   000000        XF$OFF= 000200        X$USR   000022
L.VER   000015        R.LNK   000000        S.LOA   000044        XF$ORI= 000002        X$WAQ   000062
L.WIND  000040        R.NAM   000002        S.PSS   000022        XF$OTI= 000001        X$WSZ   000020
L.XMTQ  000130        SF$BLK= 000001        S.SEC   000040        XF$TIL= 000010        $CAT5   000442RG
MA.CI = 000040        SF$INC= 000002        S.TER   000042        XF$TIR= 000004        $CFERR  000062RG
MA.DA = 000000        SF.CIR= 000010        T$FLAG  000044        XT$CHN= 000100        $CFER1  000106RG
MA.IL = 000020        SF.DEV= 000020        T$LIF   000013        XT$DGR= 000002        $C5TA   000244RG
MC.CC = 000040        SF.DIA= 004000        T$LIFL  000013        XT$FAS= 000200        $C5TA1  000234RG
MC.CI = 000020        SF.DPA= 000100        T$LIFD  000013        XT$INC= 000004        $DIV  = ****** GX
MC.DC = 000100        SF.DPC= 000200        T$LIFS  000013        XT$OUC= 000010        $DVNAM= ****** GX
MC.DI = 000060        SF.DUM= 010000        T$LIN   000000        XT$PIC= 000001        $ERDPB= ****** GX
MC.NO = 000000        SF.HAD= 000004        T$LIPS  000006        X$ABQ   000076        $EXSTS= ****** GX
MC.RC = 000140        SF.HST= 000002        T$LLD   000012        X$ALF   000055        $FLAGS= ****** GX
MD.BM = 000040        SF.LOA= 002000        T$LLDC  000045        X$AUC   000056        $FNPRO  000634RG
MD.EM = 000100        SF.PH3= 020000        T$LLDL  000012        X$CLEN= 000025        $FNSLT  000730RG
MD.ILS= 000040        SF.PSS= 000040        T$LLDO  000012        X$CTIM  000060        $FNSTA  001172RG
MD.IM = 000020        SF.SEC= 000400        T$LLDS  000012        X$DIAG  000003        $MUL  = ****** GX
MF.ACK= 000004        SF.TER= 001000        T$LLEN  000046        X$DTE   000024        $MVASC  000164RG
MF.CTL= 000010        ST$CC = 000004        T$LOPR  000002        X$FLG   000014        $PBLK = ****** GX
MF.DAT= 000000        ST$CIR= 000006        T$LTCL  000024        X$GLEN  000106        $PRINT  000000RG
NC.FM0= 000000        ST$CIS= 000002        T$LTIM  000026        X$LCN   000026        $PRNAM= ****** GX
NC.FM1= 000001        ST$DAT= 000010        T$LTPR  000014        X$LEN   000106        $QINS   000200RG
NC.FM2= 000002        ST$DIP= 000012        T$LTPS  000020        X$LENX= 000110 G      $QINSN  000216RG
NS$DON= 000000        ST$PND= 000014        T$NAPL  000004        X$MOWN  000023        $SAVAL= ****** GX
NS$SDI= 000002        S$ADSZ  000007        T$NFE   000000        X$NLRE  000052        $SAVRG= ****** GX
NS$WDC= 000004        S$CTL   000002        T$NLEN  000010        X$NNRE  000054        $STAN = ****** GX
N$$SLI= 000001        S$DTE   000020        T$NNUL  000002        X$NPL   000006        $TIDPB= ****** GX
OF.PRO= 000040        S$FLG   000010        T$NOPL  000006        X$NPR   000007        $TMLST= ****** GX
OF.RLU= 000100        S$LEN   000032        T$NRNI  000042        X$NRBY  000032        $UPBLK  001244RG
OF.SMC= 000200        S$LENX= 000034 G      T$NRPL  000005        X$NRPK  000042        $WILD = ****** GX
O.FLG   000003        S$LFT   000012        T$NRUL  000007        X$NRRE  000053        $$SHFT= 000001
O.LEN   000012        S$LNK   000000        T$NVR   000001        X$NTBY  000036        $$$R  = 000010
O.LENX= 000014 G      S$OWNR  000030        T$RPRI  000040        X$NTPK  000046        .CAT5   000432RG
O.LNK   000000        S$PKSZ  000004        T$SVC   000034        X$PKSZ  000016        .CAT5E  000422RG
O.MXC   000004        S$RCL   000013        T$T5    000030        X$PR    000011        .FNSLT  000726RG
O.NAM   000006        S$STA   000003        T$T6    000032

. ABS.  000150    000   (RW,I,GBL,ABS,OVR)
        001454    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics
```

```
    667                                             ;*****
    668                                             ; FLAGS BITS
    669                                             ;               LOW BYTE SAME AS PROCESS FLAGS
    670                                             ;
    671            000001                                 FL.DDM == 1             ; PROCESS IS A DDM
    672            000002                                 FL.DLC == 2             ; PROCESS IS A DLC
    673            000004                                 FL.LLC == 4             ; PROCESS IS AN LLC
    674            000010                                 FL.MFL == 10            ; PROCESS IS MARKED FOR LOAD
    675            000020                                 FL.KMX == 20            ; PROCESS IS A KMC MULTIPLEXER
    676            000040                                 FL.MUX == 40            ; PROCESS IS A MUX DEVICE
    677            000100                                 FL.LMC == 100           ; PROCESS HAS MICRO-CODE
    678            000200                                 FL.TIM == 200           ; PROCESS NEEDS TIMER SUPPORT
    679            000200                                 FL.DLM == 200           ; CIRCUIT IS DATALINK MAPPING CIRCUIT
    680            000400                                 FL.MTP == 400           ; LINE IS MULTI-POINT
    681            001000                                 FL.SLV == 1000          ; LINE IS A SLAVE
    682            001000                                 FL.CRE == 1000          ; CIRCUIT NEEDS TO BE CREATED
    683            002000                                 FL.X25 == 2000          ; CIRCUIT IS AN X25 CIRCUIT
    684            002000                                 FL.ADD == 2000          ; NODE ADDRESS GIVEN IN COMMAND
    685            004000                                 FL.PVC == 4000          ; CIRCUIT IS A PVC
    686            004000                                 FL.NAM == 4000          ; NODE NAME GIVEN IN COMMAND
    687            010000                                 FL.TRI == 10000         ; TRIBUTARY INFORMATION NEEDED
    688            020000                                 FL.TRB == 20000         ; PRINT TRIBUTARY ADDRESS
    689            040000                                 FL.SNA == 40000         ; LINE BELONGS TO SNA GATEWAY
```

```
                                                                 E 2
CFEIMP - SYMBOL DEFINITIONS AND MACRO V05.03b Saturday 29-Jun-85 04:55  Page 13-2
Symbol table

LUN4   = 000004     OC.NWC= 000240 G    OM.STA= 000001 G    RN.SDV= 000010 G    S$LNAM  000004 G
LUN5   = 000005     OC.PVC= 000060 G    ON.ADD= 000001 G    RN.SLO= 002000 G    S$LRPR  000020 G
L$CNT   000002 G    OC.RPR= 040000 G    ON.NAM= 000001 G    RN.SNV= 000020 G    S$LTYP  000014 G
L$LCTM  000016 G    OC.RTM= 000400 G    OO.COP= 000400 G    RN.SPA= 001000 G    S$LUNT  000011 G
L$LDEV  000012 G    OC.SLT= 046002 G    OO.NAM= 002000 G    RN.TLO= 004000 G    S$TADD  000005 G
L$LEXT  000015 G    OC.STA= 000002 G    OO.USR= 000200 G    RN.XXX= 077774 G    S$TAPR  000010 G
L$LFLG  000006 G    OC.STN= 010204 G    OO.VFY= 001000 G    R$DADR  000010 G    S$TCST  000007 G
L$LLEN= 000020 G    OC.SVC= 101400 G    OP.NCT= 000004 G    R$DLEN= 000022 G    S$TFLG  000006 G
L$LLIN  000011 G    OC.TRI= 000200 G    OP.NLT= 000002 G    R$DNAM  000004 G    S$THTM  000012 G
L$LNAM  000004 G    OC.UNT= 010204 G    OP.STA= 000001 G    R.VAR  = ****** GX  S$TLEN= 000016 G
L$LNLT  000014 G    OC.USG= 020000 G    OS.BLK= 001000 G    SF.ENA= 000100 G    S$TSTA  000004 G
L$LPRI  000010 G    OC.X25= 101570 G    OS.BUF= 000370 G    SF.MFL= 000040 G    S$VDTE  000022 G
L$NAM   000000 G    OE.ACO= 040000 G    OS.BYT= 000400 G    SF.SER= 000010 G    S$VFLG  000014 G
L$TRI   000006 G    OE.ADD= 000002 G    OS.CCB= 000010 G    SF.SVC= 000020 G    S$VLEN= 000036 G
L$UNT   000004 G    OE.AHO= 020000 G    OS.LOC= 000001 G    SL$BRA= 010000 G    S$VNAM  000004 G
MAXARE= 000077 G    OE.AMA= 010000 G    OS.NAM= 000002 G    SL$CTM= 000400 G    S$VOWN  000034 G
MAXCST= 001776 G    OE.HOS= 001000 G    OS.PAR= 000004 G    SL$HTM= 002000 G    S$VPSZ  000010 G
MAXHOP= 000036 G    OE.ID  = 000004 G   OS.PRT= 001407 G    SL$RPR= 004000 G    S$VRCL  000016 G
MAXNN  = 001777 G   OE.LNK= 000010 G    OS.RBN= 000020 G    SL$TYP= 001000 G    S$VTIM  000020 G
MXCSTL= 000031 G    OE.LV2= 070000 G    OS.RBS= 000040 G    SP.ACC= 000001 G    S$VWSZ  000012 G
NF$ACC= 100000 G    OE.MXC= 000020 G    OS.SBN= 000100 G    SP.CIR= 000004 G    S.ANM1= 000014
NF$EVT= 040000 G    OE.MXH= 000040 G    OS.THR= 000200 G    SP.DAT= 020000 G    S.ANM2= 000005
NF.ACC= 000001 G    OE.NAM= 000001 G    PV$OWN= 000400 G    SP.KNA= 000002 G    S.BFHD= 000020
NF.BCP= 000040 G    OE.NCO= 000200 G    PV$PSZ= 001000 G    SP.KPU= 001000 G    S.ETEF= 000002
NF.END= 000200 G    OE.NN  = 000100 G   PV$WSZ= 002000 G    SP.MLU= 002000 G    S.FATT= 000016
NF.EVT= 000002 G    OE.NOD= 001007 G    P$SFLG  000012 G    SP.MOD= 000040 G    S.FDB  = 000140
NF.LLI= 000004 G    OE.NSP= 000610 G    P$SLEN= 000020 G    SP.PLU= 000020 G    S.FNAM= 000006
NF.LV2= 000100 G    OE.ROU= 072160 G    P$SNAM  000004 G    SP.PSW= 000200 G    S.FNB  = 000036
NF.SLI= 000400 G    OE.RTM= 002000 G    P$SPOR  000010 G    SP.SID= 010000 G    S.FNBW= 000017
NF.SMC= 000020 G    OE.SUB= 004000 G    P$STHI  000016 G    SP.SLU= 000010 G    S.FNTY= 000004
NF.SWP= 000010 G    OE.VFY= 000400 G    P$STLO  000014 G    SP.SPU= 000400 G    S.FTYP= 000002
NO$EXA= 000400 G    OL.BAU= 000040 G    P$VCT   000012 G    SP.UID= 000100 G    S.NFEN= 000020
NO$HOA= 001000 G    OL.BSZ= 020000 G    P$VFLG  000022 G    SP.XID= 004000 G    UN$CST= 001000 G
N.ANM1= 000002      OL.CNT= 000624 G    P$VLCN  000010 G    ST$HTM= 000400 G    UN$DPR= 002000 G
N.ANM2= 000024      OL.CSR= 000200 G    P$VLEN= 000024 G    SV$DTE= 002000 G    UN$PCH= 004000 G
N.DID  = 000024     OL.CTM= 000001 G    P$VNAM  000004 G    SV$PSZ= 000400 G    UN$SCS= 000400 G
N.DVNM= 000032      OL.DPR= 001000 G    P$VOWN  000020 G    SV$WSZ= 001000 G    U$NCSR  000012 G
N.FID  = 000000     OL.HBT= 002000 G    P$VPSZ  000014 G    SW.ENA= 000004 G    U$NCST  000014 G
N.FNAM= 000006      OL.NWC= 000320 G    P$VSTA  000023 G    SW.FUL= 000010 G    U$NCW0  000006 G
N.FTYP= 000014      OL.PLX= 000010 G    P$VWSZ  000016 G    SW.INC= 002000 G    U$NCW1  000010 G
N.FVER= 000016      OL.PRI= 000004 G    Q.IOAE= 000012 G    SW.LOA= 000002 G    U$NDPR  000016 G
N.NEXT= 000022      OL.RTC= 004000 G    Q.IOEF= 000006      SW.LON= 000200 G    U$NLEN= 000026 G
N.STAT= 000020      OL.RIT= 040000 G    Q.IOFN= 000002 G    SW.OUT= 001000 G    U$NPCH  000024 G
N.UNIT= 000034      OL.SEC= 000100 G    Q.IOLU= 000004      SW.PUR= 000400 G    U$NUNT  000004 G
OC.APR= 010000 G    OL.SLT= 000003 G    Q.IOPL= 000014 G    SW.RLU= 000020 G    WC.CIR= 000017 G
OC.CHN= 000040 G    OL.STA= 000002 G    Q.IOPR= 000007      SW.SDI= 010000 G    WC.CNT= 000002 G
OC.CNT= 000000 G    OL.STN= 000000 G    Q.IOSB= 000010 G    SW.SEN= 004000 G    WC.EVE= 000001 G
OC.CST= 000004 G    OL.UNT= 001150 G    RE$ARE= 000400 G    SW.SLT= 014000 G    WC.LIN= 000017 G
OC.CTM= 000001 G    OL.URM= 000400 G    RN.DCT= 000100 G    SW.SMC= 000040 G    WC.LOG= 000002 G
OC.DTE= 000020 G    OL.VEC= 000020 G    RN.DIA= 020000 G    SW.TOP= 000001 G    WC.NAM= 000001 G
OC.EXE= 002000 G    OL.WSZ= 010000 G    RN.DMP= 040000 G    SW.VFY= 000100 G    WC.NOD= 000001 G
OC.HTM= 002000 G    OL.X25= 076000 G    RN.DPA= 000040 G    S$LBRA  000022 G    WC.OBJ= 000001 G
OC.MBL= 000100 G    OM.CON= 000004 G    RN.HAR= 000400 G    S$LCNT  000010 G    WC.PRO= 000001 G
OC.MRC= 001000 G    OM.EVE= 000020 G    RN.HOS= 000004 G    S$LCTT  000012 G    WC.TRI= 000010 G
OC.MWI= 000010 G    OM.FIL= 000010 G    RN.LOD= 010000 G    S$LFLG  000006 G    WC.UNT= 000004 G
OC.MXR= 004000 G    OM.KEV= 000040 G    RN.NAM= 000002 G    S$LHTM  000016 G    W.TSEF= 000002
OC.NUM= 100000 G    OM.MON= 000002 G    RN.SCI= 000200 G    S$LLEN= 000024 G    XA.DST= 000001 G
```

```
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL   VALUE            REFERENCES

XS.CTM  = 000004  G       #11-984      11-1006
XS.CUG  = 000100  G       #11-989      11-1011
XS.DST  = 000001  G       #11-982      11-998      11-1004
XS.DTE  = 000200  G       #11-990      11-1012
XS.KND  = 000002  G       #11-983      11-999      11-1005
XS.MXC  = 000010  G       #11-985      11-1007
XS.OBI  = 001000  G       #11-992      11-1014
XS.OBJ  = 000400  G       #11-991      11-1013
XS.PRI  = 002000  G       #11-993      11-1015
XS.STA  = 000020  G       #11-979
XS.ST1  = 000040  G       #11-980
XS.SUB  = 004000  G       #11-994      11-1016
X$2BSZ    000014  G       #9-482
X$2HBT    000022  G       #9-485
X$2LEN  = 000024  G       #9-486
X$2LIN    000004  G       #9-479
X$2RTC    000016  G       #9-483
X$2RTT    000020  G       #9-484
X$2WSZ    000012  G       #9-481
X$3CT     000012  G       #9-596
X$3DFP    000004  G       #9-592
X$3DW     000010  G       #9-594
X$3KM     000016  G       #9-600
X$3KT     000013  G       #9-597
X$3LEN  = 000021  G       #9-603
X$3MW     000011  G       #9-595
X$3MXP    000006  G       #9-593
X$3RM     000017  G       #9-601
X$3RT     000014  G       #9-598
X$3SM     000020  G       #9-602
X$3ST     000015  G       #9-599
X$9CT     000006  G       #9-580
X$9LEN  = 000010  G       #9-581
X$9MXC    000004  G       #9-579
X9.CLM  = 000020  G       #11-1009
X9.CLV  = 000040  G       #11-1010
X9.CTM  = 000004  G       #11-1006
X9.CUG  = 000100  G       #11-1011
X9.DST  = 000001  G       #11-1004
X9.DTE  = 000200  G       #11-1012
X9.KND  = 000002  G       #11-1005
X9.MXC  = 000010  G       #11-1007
X9.OBI  = 001000  G       #11-1014
X9.OBJ  = 000400  G       #11-1013
X9.PRI  = 002000  G       #11-1015
X9.SUB  = 004000  G       #11-1016
ZF.COU  = 001000  G       #12-1058
ZF.DDM  = 000001  G       #12-1049
ZF.DIA  = 020000  G       #12-1063
ZF.DLC  = 000002  G       #12-1050
ZF.DVP  = 040000  G       #12-1064
ZF.INI  = 100000  G       #12-1065
```

```
                                                                      E  4
CFEINI      CREATED BY  MACRO  ON 29-JUN-85 AT 04:56       PAGE 1

SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL   VALUE              REFERENCES

CL.CMT   = ******  GX        5-110      5-112
ERRFIL   = ******  GX        5-126
ERRGCL   = ******  GX        5-117
EX$SUC   = ******  GX        5-104
IE.EOF   = ******  GX        5-115
PAR$$$   = 000027           #5-125      5-125     #5-136      5-136     #5-137      5-137
PRMPT      000000 R         #4-85       5-108
$BUFF    = ******  GX        5-120
$BYTE    = ******  GX       *5-133
$CFERR   = ******  GX        5-117      5-126
$CFINI     000024 RG        #5-103
$CLFLG   = ******  GX       *5-110     *5-112
$CLIOS   = ******  GX        5-109
$CLPMT   = ******  GX        5-107     *5-108     *5-113
$EXSTS   = ******  GX       *5-104
$GCLEP   = ******  GX        5-111
$HIVBN   = ******  GX       *5-131
$IFDB    = ******  GX        5-125      5-125      5-129      5-136      5-136
$IOPEN   = ******  GX        5-121
$LOVBN   = ******  GX       *5-132
$OFDB    = ******  GX        5-137      5-137
$OPEN    = ******  GX        5-123
$TMSTB   = ******  GX        5-134
$XXINI   = ******  GX        5-105
.CLOSE   = ******  G         5-125      5-125      5-136      5-136
.DELET   = ******  G         5-137      5-137
.FINIT   = ******  G         5-106      5-106
.MARK    = ******  GX        5-130
```

```
                                          ;+
                                          ; *** - $IOPEN - OPEN THE INPUT FILE
                                          ;
                                          ; INPUTS:
                                          ;         R1 = ADDRESS OF FILE-SPEC
                                          ;         R2 = LENGTH OF FILE-SPEC
                                          ;
                                          ; OUTPUTS:
                                          ;         C-BIT = SUCCESS/FAILURE
                                          ;         R0 = FDB ADDRESS
                                          ;         R1,R2 = PRESERVED
                                          ;         IF SUCCESS: FILE IS OPENED FOR UPDATE
                                          ;
                                          ;-
000354                                    $IOPEN::
000354  012700  000300'                           MOV     #CSBLK,R0       ; POINT R0 AT CSI CONTROL BLOCK
000360  010260  000002                            MOV     R2,C.CMLD(R0)   ; SET LENGTH OF FILE-SPEC STRING
000364  010160  000004                            MOV     R1,C.CMLD+2(R0) ; AND ADDRESS OF FILE-SPEC
000370                                            CSI$1                   ; PARSE FILESTRING
000374  103404                                    BCS     10$             ; IF CS, ERROR
000376                                            CSI$2   ,OUTPUT         ; GET OUTPUT FILE SPEC
000406  012700  000000G           10$:            MOV     #$IFDB,R0       ; POINT AT FDB
000412  112760  000000G 000052                    MOVB    #IE.BNM,F.ERR(R0) ; ASSUME BAD FILENAME
000420  103425                                    BCS     20$             ; IF CS, RETURN WITH ERROR CODE
000422                                            FDOP$R  ,,#CSBLK+C.DSDS ; POINT AT PARSED FILE-SPEC
000430                                            OPEN$R                  ; OPEN INPUT FILE
000442  103414                                    BCS     20$             ; IF CS, RETURN
                                          ;
                                          ; SAVE OWNER INFORMATION AND PROTECTION OF INPUT FILE
                                          ;
000444  116067  000042  177332                    MOVB    F.LUN(R0),RATDPB+Q.IOLU ; GET LUN INTO DPB
000452                                            DIR$    #RATDPB         ; TRY TO READ ATTRIBUTES
000460  103403                                    BCS     15$             ; IF CS, DIRECTIVE FAILED
000462  105767  177342                            TSTB    IOSB            ; ELSE, CHECK STATUS
000466  100002                                    BPL     20$             ; IF PL, OK
000470  105067  000000G           15$:            CLRB    $FIATR          ; ELSE CLEAR TO PREVENT LATER WRITE
000474  000207                    20$:            RETURN                  ; RETURN TO CALLER
```

```
                                                .SBTTL  $CFPRG - PERFORM PURGE COMMAND
                                        ;+
                                        ; *** - $CFPRG - PERFORM PURGE COMMAND
                                        ;
                                        ; INPUT:
                                        ;       $COMP = THE COMPONENT CODE
                                        ;
                                        ; OUTPUT:
                                        ;       CONTROL IS PASSED TO THE PROPER ROUTINE
                                        ;
                                        ;
                                        ;-

 000214                                 $CFPRG::
 000214 016700  000000G                         MOV     $COMP,R0        ; GET THE COMPONENT CODE
 000220 006300                                  ASL     R0              ; CONVERT TO A WORD INDEX
 000222 000170  000144'                         JMP     @DSPTBL-2(R0)   ; DISPATCH TO PROPER ROUTINE
```

Table of contents

```
     4-   54     MACROS AND STORAGE
     4-   79     KEYWORDS
     4-  281     COMMANDS
     5-  302     LIST COMPONENTS
     6-  432     PURGE COMPONENTS
     6-  656     DEFINE LOGGING MONITOR
     6-  692     SUB-EXPRESSIONS FOR EVENT ID
     6-  737     SUB-EXPRESSION TO PARSE LOGGING MONITOR
     7-  758     DEFINE NODE COMMAND
     8- 1081     DEFINE PROCESS COMMAND
     8- 1133     SUB-EXPRESSIONS FOR PROCESS ID
     9- 1151     DEFINE CIRCUIT COMMAND
     9- 1318     DEFINE LINE COMMAND
     9- 1513     SUB-EXPRESSIONS FOR CIRCUIT/LINE ID
    10- 1590     DEFINE EXECUTOR COMMAND
    11- 1914     DEFINE OBJECT COMMAND
    12- 1986     DEFINE SYSTEM COMMAND
    13- 2137     DEFINE MODULE COMMAND
    14- 2427     UTILITY SUBEXPRESSIONS
    15- 2637     EXIT STATE
```

PURGE COMPONENTS

```
    603 000000                          STATE$  LOGOPT
    604 000000                           TRAN$  !OPEVE, $EXIT,, OM.EVE, $OPRND
    605                         ;
    606                         ; PURGE LINE OPTIONS - COUNTER TIMER
    607                         ;
    608 000000                          STATE$  PURLIN,&P$PUOP
    609 000000                           TRAN$  !PURCT, $EXIT,  .LIX25
    610                         ;
    611                         ; PURGE CIRCUIT OPTIONS
    612                         ;
    613 000000                          STATE$  PURCIR, &P$PUOP
    614 000000                           TRAN$  !PURCT, $EXIT,,         OC.CTM, $OPRND
    615 000000                           TRAN$  $LAMDA, PUROPT
    616                         ;
    617                         ; COUNTER TIMER PURGE OPTION
    618                         ;
    619 000000                          STATE$  PURCT
    620 000000                           TRAN$  &K$COU
    621 000000                          STATE$  ,&P$CTO
    622 000000                           TRAN$  &K$TIM, $EXIT
    623                         ;
    624                         ; PURGE MODULE X25-PROTOCOL OPTIONS
    625                         ;
    626 000000                          STATE$  PMXPRA, &P$PMOD
    627 000000                           TRAN$  $LAMDA
    628 000000                          STATE$  PMXPRX
    629 000000                           TRAN$  !DTE,   PMXPR1,,XP.DTE,$OPRND
    630 000000                           TRAN$  !KND,   PMXPR1,,XP.KND,$OPRND
    631 000000                           TRAN$  !CUG,   PUROPT,,XP.CUG,$OPRND
    632 000000                           TRAN$  !KNG,   PUROPT,,XP.KNG,$OPRND
    633 000000                          STATE$  PMXPR1
    634 000000                           TRAN$  !PURCT, $EXIT,, XP.CTM,$OPRND
    635 000000                           TRAN$  $LAMDA, PUROPT
    636                         ;
    637                         ; PURGE MODULE X25-ACCESS OPTIONS
    638                         ;
    639 000000                          STATE$  PMXACA, &P$PMOD
    640 000000                           TRAN$  $LAMDA, PMXDST
    641                         ;
    642                         ; PURGE MODULE X25/X29-SERVER OPTIONS
    643                         ;
    644 000000                          STATE$  PMXSRA, &P$PMOD
    645 000000                           TRAN$  $LAMDA
    646 000000                          STATE$  PMXSRV
    647 000000                           TRAN$  !PURCT, $EXIT,, XS.CTM, $OPRND
    648 000000                           TRAN$  $LAMDA
    649                         ;
    650                         ; PURGE MODULE X25/X29-SERVER, X25-ACCESS COMMON OPTIONS
    651                         ;
    652 000000                          STATE$  PMXDST
    653 000000                           TRAN$  !KNDS,  PUROPT,,XS.KND, $OPRND
    654 000000                           TRAN$  !DST,   PUROPT,,XS.DST, $OPRND
    655
    656                                 .SBTTL  DEFINE LOGGING MONITOR
    657                         ;
    658                         ; DEFINE LOGGING MONITOR
    659                         ;
```

```
                                ; COUNTER TIMER
                                ;
 000000                                 STATE$  CT
 000000                                  TRAN$  &K$COU
 000000                                 STATE$  ,&P$CTO
 000000                                  TRAN$  &K$TIM
 000000                                 STATE$  ,&P$CT
 000000                                  TRAN$  $DNUMB,$EXIT,.CICT
                                ;
                                ; HELLO TIMER
                                ;
 000000                                 STATE$  HTM
 000000                                  TRAN$  &K$HLO
 000000                                 STATE$  ,&P$HTO
 000000                                  TRAN$  &K$TIM
 000000                                 STATE$  ,&P$HTM
 000000                                  TRAN$  $DNUMB,$EXIT,.CIHTM
                                ;
                                ; MAXIMUM DATA
                                ;
 000000                                 STATE$  MBL
 000000                                  TRAN$  &K$MAX
 000000                                 STATE$  ,&P$MAX
 000000                                  TRAN$  &K$DAT
 000000                                 STATE$  ,&P$MBD
 000000                                  TRAN$  $DNUMB,$EXIT,.CIMBL
                                ;
                                ; MAXIMUM RECALLS
                                ;
 000000                                 STATE$  MRC
 000000                                  TRAN$  &K$MAX
 000000                                 STATE$  ,&P$MAX
 000000                                  TRAN$  &K$RCL
 000000                                 STATE$  ,&P$MRC
 000000                                  TRAN$  $DNUMB,$EXIT,.CIMRC
                                ;
                                ; RECALL TIMER
                                ;
 000000                                 STATE$  RTM
 000000                                  TRAN$  &K$RCL
 000000                                 STATE$  ,&P$RCO
 000000                                  TRAN$  &K$TIM
 000000                                 STATE$  ,&P$RTM
 000000                                  TRAN$  $DNUMB,$EXIT,.CIRTM
                                ;
                                ; CIRCUIT USEAGE
                                ;
 000000                                 STATE$  USG
 000000                                  TRAN$  &K$USG
 000000                                 STATE$  ,&P$USG
 000000                                  TRAN$  &K$PER,$EXIT
 000000                                  TRAN$  &K$INC,$EXIT,,SW.INC,$SWICH
 000000                                  TRAN$  &K$OUT,$EXIT,,SW.OUT,$SWICH
                                        .SBTTL  DEFINE LINE COMMAND
                                ;
                                ; DEFINE LINE COMMAND
                                ;
```

```
    1971                                    ;
    1972                                    ; LOOK FOR OBJECT ID
    1973                                    ;
    1974 000000                                     STATE$  OBJID
    1975 000000                                      TRAN$  &K$OBJ
    1976
    1977 000000                                     STATE$  ,&P$OID
    1978 000000                                      TRAN$  $DNUMB, $EXIT,  .OBNUM
    1979
    1980 000000                                     STATE$  KNOBJ
    1981 000000                                      TRAN$  &K$KNO
    1982
    1983 000000                                     STATE$  ,&P$PEN
    1984 000000                                      TRAN$  &K$OBJS,$EXIT,,        WC.OBJ, $WILD
```

```
000000                                  STATE$  MWI
000000                                   TRAN$  &K$MAX
000000                                  STATE$  ,       &P$MAX
000000                                   TRAN$  &K$WIN
000000                                  STATE$  ,       &P$MWI
000000                                   TRAN$  $DNUMB, $EXIT,   .PRMWI
                                ;;;;
                                ;;;; MAXIMUM DATA
                                ;;;;
                                ;;;     STATE$  MDA
                                ;;;      TRAN$  &K$MAX
                                ;;;     STATE$  ,       &P$MAX
                                ;;;      TRAN$  &K$DAT
                                ;;;     STATE$  ,       &P$MBD
                                ;;;      TRAN$  $DNUMB, $EXIT,   .MOMDA
                                ;;;      TRAN$  &K$DAT
                                ;;;


                                ;
                                ; "MODULE SNA-PROTOCOL"
                                ;
000000                                  STATE$  MODSPR
000000                                   TRAN$  &K$MOD,MSPR1
000000                                   TRAN$  $LAMDA,MSPR2

000000                                  STATE$  MSPR1,&P$MOD
000000                                   TRAN$  $LAMDA

000000                                  STATE$  MSPR2
000000                                   TRAN$  &K$SNA,,$BL.MN

000000                                  STATE$
000000                                   TRAN$  <'->

000000                                  STATE$
000000                                   TRAN$  &K$PRT,$EXIT,$BL.A3
000000                                   TRAN$  $LAMDA,,.BL.AU

```

E 12

CFEROT - ROOT SEGMENT FOR CFE   MACRO V05.03b Saturday 29-Jun-85 05:13
Table of contents

5- 107 $CFE - CFE ROOT SEGMENT

F 12

```
                                        .SBTTL  .RQCB   REQUEST CORE BLOCK
                                        .SBTTL  $RLBF   RELEASE CORE BLOCK
                                ;
                                ;
                                ; BEST/FIRST FIT DYNAMIC CORE ALLOCATION ROUTINES
                                ;
                                ; DATA STRUCTURES
                                ;
                                ; FREE BLOCK LIST HEADER:
                                ;
                                ;       FREEHD: .WORD   NEXT    ;POINTER TO NEXT FREE BLOCK OR 0
                                ;               .WORD   0       ;ALWAYS ZERO
                                ;
                                ; FREE BLOCK:
                                ;
                                ;       FREEBK: .WORD   NEXT    ;POINTER TO NEXT OR 0
                                ;               .WORD   SIZE    ;SIZE OF BLOCK IN BYTES
                                ;
                                ; NOTE: THE FREE BLOCK LIST IS ORDERED BY THE CORE ADDRESS OF THE BLOCKS
                                ;       THEMSELVES. THIS IS SO GARBAGE COLLECTION AND BLOCK AGGLOMERATIO
                                ;       CAN BE PERFORMED WITH LITTLE OVERHEAD.
                                ;
                                ;+
                                ; **-$RLBF-RELEASE CORE BLOCK
                                ;
                                ; THIS SUBROUTINE IS CALLED TO RELEASE A CORE BLOCK TO THE FREE LIST.
                                ; THE FREE LIST IS SEARCHED UNTIL THE PROPER SLOT IS FOUND AND THE
                                ; BLOCK IS MERGED INTO THE FREE LIST. IF THE BLOCK BEING RELEASED
                                ; CAN BE AGGLOMERATED WITH A BLOCK ALREADY IN THE FREE LIST, THEN THE
                                ; BLOCKS ARE AGGLOMERATED AND THE NEW BLOCK IS MERGED INTO THE FREE
                                ; LIST.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0=ADDRESS OF FREE BLOCK LIST HEAD.
                                ;       R1=SIZE OF BLOCK TO RELEASE IN BYTES.
                                ;       R2=ADDRESS OF BLOCK TO RELEASE.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ;-

 000102  004567  000000G        $RLBF:: JSR     R5,$SAVRG               ;SAVE NONVOLITILE REGISTERS
 000106  062701  000003                 ADD     #3,R1           ;ROUND UP TO NEXT 2 WORD BOUNDRY
 000112  042701  000003                 BIC     #3,R1           ;CLEAR EXCESS BITS
 000116  001432                         BEQ     5$              ;IF EQ VERGESSEN!!!
 000120  011003                 1$:     MOV     (R0),R3         ;GET ADDRESS OF NEXT IN LIST
 000122  001404                         BEQ     2$              ;IF EQ END OF CHAIN
 000124  020203                         CMP     R2,R3           ;COMPARE ADDRESSES
 000126  103402                         BLO     2$              ;IF LO FOUND SLOT
 000130  010300                         MOV     R3,R0           ;SET NEW PREVIOUS ADDRESS
 000132  000772                         BR      1$              ;GO AGAIN
                                ;
                                ; SEE IF BLOCKS CAN BE MERGED IN NEW + OLD ORDER.
                                ;
 000134  010312                 2$:     MOV     R3,(R2)         ;ASSUME NO AGGLOMERATION
 000136  010204                         MOV     R2,R4           ;CALCULATE ADDRESS OF NEXT BLOCK
```

```
                                               .SBTTL  $T1ALC - ALLOCATE TEMPLATES
                                     ;+
                                     ; *** - $T1ALC - ALLOCATE TEMPLATES OR EXIT ON ERROR
                                     ;
                                     ; INPUTS:
                                     ;         R1 = LENGTH OF ALLOCATION REQUEST
                                     ;
                                     ; OUTPUTS:
                                     ;         R0 = ADDRESS OF ALLOCATED BUFFER
                                     ;
                                     ;-

000170                               $T1ALC::
000170  012700  000000G                        MOV     #$XXAVL,R0              ; GET ADDRESS OF END OF TASK LISTHEAD
000174  010146                                 MOV     R1,-(SP)                ; PUSH LENGTH
000176  004767  000000G                        CALL    $RQBF                   ; TRY TO ALLOCATE BLOCK
000202  103006                                 BCC     10$                     ; IF CC, GOT ONE
000204                                         ERROR$  #ERRALL                 ; ELSE, ALLOCATION FAILURE
000214  000167  000000G                        JMP     .EXIT                   ; AND EXIT IMMEDIATELY
000220  010001                       10$:      MOV     R0,R1                   ; COPY POINTER
000222  105021                       20$:      CLRB    (R1)+                   ; CLEAR ONE BYTE
000224  005316                                 DEC     (SP)                    ; COUNTDOWN
000226  003375                                 BGT     20$                     ; UNTIL BLOCK IS ZEROED
000230  005726                                 TST     (SP)+                   ; CLEAN STACK
000232  000207                                 RETURN

        000001                                 .END
```

```
                                        .SBTTL  $QINS - INSERT ENTRY AT END OF QUEUE
                                        .SBTTL  $QINSN - INSERT ENTRY INTO MIDDLE OF QUEUE
                                ;+
                                ; *** - $QINS - INSERT ENTRY AT END OF QUEUE
                                ; *** - $QINSN - INSERT ENTRY INTO MIDDLE OF QUEUE
                                ;
                                ; INPUT:
                                ;       R0 = ADDRESS OF ENTRY TO INSERT INTO QUEUE
                                ;       R1 = ADDRESS OF DOUBLE WORD LISTHEAD ($QINS) OR PREVIOUS ENTRY ($QINSN)
                                ;
                                ; REGISTERS ALTERED:
                                ;       R2
                                ;
                                ;
                                ;-

000200                          $QINS::
000200  016102  000002                  MOV     2(R1),R2        ; GET LAST ENTRY IN QUEUE
000204  010061  000002                  MOV     R0,2(R1)        ; SET NEW ENTRY AS LAST
000210  010012                          MOV     R0,(R2)         ; LINK NEW ENTRY INTO CHAIN
000212  005010                          CLR     (R0)            ; CLEAR POINTER TO NEXT
000214  000207                          RETURN

000216                          $QINSN::
000216  011110                          MOV     (R1),(R0)       ; SET NEW NEXT ENTRY
000220  001002                          BNE     10$             ; IF NE, NO NEW LAST
000222  010067  000002G                 MOV     R0,$TMLST+2     ; ELSE, SET NEW LAST
000226  010011                  10$:    MOV     R0,(R1)         ; INSERT NEW ENTRY INTO LIST
000230  000207                          RETURN
```

```
Work  file  reads: 0
Work  file writes: 0
Size of work file: 16287 Words  ( 64 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:26.16
SY:CFESUB.V2,[132,134]CFESUB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFESUB
```

```
    691
    692                                          ;*****
    693                                          ; COMMAND CODES
    694                                          ;
    695        000001                                   CM.DEF == 1             ; "DEFINE" COMMAND
    696        000002                                   CM.LIS == 2             ; "LIST" COMMAND
    697        000003                                   CM.HLP == 3             ; "HELP" COMMAND
    698        000004                                   CM.EXI == 4             ; "EXIT" COMMAND
    699        000005                                   CM.KIL == 5             ; "KILL" COMMAND
    700        000006                                   CM.PUR == 6             ; "PURGE" COMMAND
    701
    702                                          ;
    703                                          ;*****
    704                                          ; COMPONENT CODES
    705                                          ;
    706        000001                                   CO.NOD == 1             ; "NODE"
    707        000001                                   CO.KNN == 1             ; "KNOWN NODES"
    708        000002                                   CO.PRO == 2             ; "PROCESS"
    709        000002                                   CO.KNP == 2             ; "KNOWN PROCESSES"
    710        000003                                   CO.EXE == 3             ; "EXECUTOR"
    711        000004                                   CO.SYS == 4             ; "SYSTEM"
    712        000005                                   CO.OBJ == 5             ; "OBJECT"
    713        000005                                   CO.KNO == 5             ; "KNOWN OBJECTS
    714        000006                                   CO.LIN == 6             ; "LINE"
    715        000006                                   CO.KNL == 6             ; "KNOWN LINES"
    716        000007                                   CO.LOG == 7             ; "LOGGING "
    717        000007                                   CO.KLG == 7             ; "KNOWN LOGGING"
    718        000010                                   CO.CIR == 10            ; "CIRCUIT"
    719        000011                                   CO.KNC == 11            ; "KNOWN CIRCUITS"
    720        000012                                   CO.MXP == 12            ; "MODULE X25-PROTOCOL"
    721        000013                                   CO.MXS == 13            ; "MODULE X25-SERVER"
    722        000014                                   CO.MAC == 14            ; "MODULE X25-ACCESS"
    723        000015                                   CO.MX9 == 15            ; "MODULE X29-SERVER"
    724        000016                                   CO.MSP == 16            ; "MODULE SNA-PROTOCOL"
    725                                          ;
    726                                          ;*****
    727                                          ; SWITCH BIT DEFINITIONS
    728                                          ;
    729        000001                                   SW.TOP == 1             ; TOP DOWN LOADING
    730        000002                                   SW.LOA == 2             ; AUTOMATIC LOAD
    731        000004                                   SW.ENA == 4             ; AUTOMATIC ENABLE
    732        000010                                   SW.FUL == 10            ; FULL DUPLEX LINE
    733        000020                                   SW.RLU == 20            ; OBJECT TASK IS TO RUN ON LOGIN UIC
    734        000040                                   SW.SMC == 40            ; OBJECT TASK SHOULD SPAWN MULT. COPIES
    735        000100                                   SW.VFY == 100           ; LOCAL VERIFICATION IS ON
    736        000200                                   SW.LON == 200           ; LOGGING STATE IS ON
    737        000400                                   SW.PUR == 400           ; EXIT AND PURGE COMMAND
    738        001000                                   SW.OUT == 1000          ; USAGE IS OUTGOING     (PSI, SVC ONLY)
    739        002000                                   SW.INC == 2000          ; USAGE IS INCOMING     (PSI, SVC ONLY)
    740        004000                                   SW.SEN == 4000          ; CIRCUIT SERVICE ENABLED
    741        010000                                   SW.SDI == 10000         ; CIRCUIT SERVICE DISABLED
    742
    743        014000                                       SW.SLT == SW.SEN!SW.SDI ; SLT SWITCHES
    744                                          ;
    745                                          ;*****
    746                                          ; WILD CARD BIT DEFINITIONS
    747                                          ;
```

```
                                                          F 2
CFEIMP - SYMBOL DEFINITIONS AND MACRO V05.03b Saturday 29-Jun-85 04:55  Page 13-3
Symbol table

XA.KND= 000002 G    X$3RM   000017 G        $CLAST  000172RG  003 $HBT   = 001602RG      $RNSDV= 001554RG
XA.NUM= 000004 G    X$3RT   000014 G        $CLBUF= 000310RG  003 $HIVBN  001472RG      $RNSLO= 001620RG
XP.BIL= 000004 G    X$3SM   000020 G        $CLCSI  000212RG  003 $HOST = 001560RG      $RNSPA= 001604RG
XP.CHN= 000001 G    X$3ST   000015 G        $CLEVL  000206RG  003 $HTM  = 001620RG      $RNTLO= 001657RG
XP.CLT= 000002 G    X$9CT   000006 G        $CLFDB  000266RG  003 $ID     001622RG      $RNXXX= 002053RG
XP.COU= 000002 G    X$9LEN= 000010 G        $CLFLG  000464RG  003 $IFDB   000142RG      $RST    001574RG
XP.CTM= 000001 G    X$9MXC  000004 G        $CLIOS= 000132RG  003 $INFO   001464RG      $RTC  = 001616RG
XP.CUG= 040000 G    X9.CLM= 000020 G        $CLM    001726RG      $LINKS  001562RG      $RTM    001722RG
XP.DFB= 000004 G    X9.CLV= 000040 G        $CLMCR  000000RG  003 $LOVBN  001474RG      $SACNM= 001734RG
XP.DFW= 000010 G    X9.CTM= 000004 G        $CLPMT  000466RG  003 $LSNAD  001436RG      $SCIRC= 001724RG
XP.DTA= 000001 G    X9.CUG= 000100 G        $CLPRM  000472R   003 $LSTMP  001434RG      $SCSR   001560RG
XP.DTE= 010000 G    X9.DST= 000001 G        $CLQIO  000122RG  003 $MAXC = 001550RG      $SDAT = 001550RG
XP.KND= 020000 G    X9.DTE= 000200 G        $CLRTN  000000R       $MAXH = 001552RG      $SDBNM  001526RG
XP.KNG= 100000 G    X9.KND= 000002 G        $CLSBK  000176RG  003 $MBL  = 001564RG      $SLUCT= 001622RG
XP.LIN= 000004 G    X9.MXC= 000010 G        $CLSEF  000156RG  003 $MCI  = 001542RG      $SLUS = 001624RG
XP.LV3= 003777 G    X9.OBI= 001000 G        $CLSIZ= 000306RG  003 $MRC  = 001560RG      $SMOD = 001754RG
XP.MCI= 000010 G    X9.OBJ= 000400 G        $CLSRV  000470RG  003 $MRE  = 001550RG      $SNAPT  001426RG
XP.MRE= 000100 G    X9.PRI= 002000 G        $CLT  = 001562RG      $MWI  = 001556RG      $SNLU = 001744RG
XP.MXB= 000020 G    X9.SUB= 004000 G        $CLTBF  000202RG  003 $MXC  = 001560RG      $SNSID= 001746RG
XP.MXC= 000040 G    ZF.COU= 001000 G        $CLTSZ  000204RG  003 $MXNCT= 000100 G      $SPARE= 001734RG
XP.MXR= 000200 G    ZF.DDM= 000001 G        $CLV    001756RG      $MXNLT= 000100 G      $SPLU = 001744RG
XP.MXW= 000400 G    ZF.DIA= 020000 G        $CLWEF  000152RG  003 $MXR  = 001570RG      $SPSW = 001774RG
XP.NNA= 004000 G    ZF.DLC= 000002 G        $CMAND  001446RG      $MXSTA= 000100 G      $SPUNM= 001724RG
XP.NUM= 000002 G    ZF.DVP= 040000 G        $CMDPT= 001450RG      $NCOU = 001574RG      $SSERN= 001750RG
XP.RET= 001000 G    ZF.INI= 100000 G        $CNFIG  001440RG      $NCT  = 001620RG      $SSSCP= 001754RG
XP.RST= 002000 G    ZF.KMX= 000020 G        $CNTNM  001534RG      $NLN    001706RG      $STADR= 001566RG
XP.STA= 000020 G    ZF.LLC= 000004 G        $COMP   001450RG      $NLT  = 001564RG      $STAN   001540RG
XP.ST1= 000040 G    ZF.LMC= 000100 G        $COPES  001574RG      $NN   = 001556RG      $SUB1   002004RG
XS.CLM= 000020 G    ZF.MAN= 004000 G        $COST   001710RG      $NNAM   001724RG      $SUB2   002006RG
XS.CLV= 000040 G    ZF.MFL= 000010 G        $CSR    001556RG      $NODAD  001546RG      $SUID = 001764RG
XS.CTM= 000004 G    ZF.MTM= 000400 G        $CT   = 001574RG      $NODNA  001542RG      $SVUPD  001430RG
XS.CUG= 000100 G    ZF.MUX= 000040 G        $CUG  = 001576RG      $NTADD  001504RG      $SWICH  001460RG
XS.DST= 000001 G    ZF.PSE= 002000 G        $DEFNB  000302RG      $NTNAM  001500RG      $THRSH  001530RG
XS.DTE= 000200 G    ZF.SLI= 010000 G        $DFD  = 001602RG      $NUM  = 001550RG      $TIDPB  000354RG
XS.KND= 000002 G    ZF.TIM= 000200 G        $DFW  = 001540RG      $OBJ    001712RG      $TMLST  001420RG
XS.MXC= 000010 G    ZF.X3P= 002000 G        $DPR  = 001570RG      $OFDB   000002RG      $TNAM   001576RG
XS.OBI= 001000 G    $APR  = 001572RG        $DST    001716RG      $OPRND  001452RG      $TNUM   001602RG
XS.OBJ= 000400 G    $ASTLV  000210RG    003 $DTE    001622RG      $OPRN2  001454RG      $UMRIN  001512RG
XS.PRI= 002000 G    $BAUD   001554RG        $DVNAM= 001532RG      $PARNA  001570RG      $UMRUP  001514RG
XS.STA= 000020 G    $BSZ  = 001564RG        $EACOS  002012RG      $PBLK   001516RG      $UNTNM  001536RG
XS.ST1= 000040 G    $BUFF   000434RG        $EAHOP  002014RG      $POLNA  001564RG      $URM  = 001562RG
XS.SUB= 004000 G    $BYT  = 001550RG        $EAMAX  002010RG      $PRI    001552RG      $VEC    001550RG
X$2BSZ  000014 G    $BYTE   001476RG        $EBEA   002020RG      $PRNAM  001532RG      $VLVL   001620RG
X$2HBT  000022 G    $BYTMN  001506RG        $EBRT   002016RG      $PSHPT  000174RG  003 $WILD   001462RG
X$2LEN= 000024 G    $BYTXT  001510RG        $EINA   002024RG      $PSIPT  001424RG      $WSZ  = 001556RG
X$2LIN  000004 G    $CALT = 001576RG        $EINC   002025RG      $RDBNM  001522RG      $$$   = 001774
X$2RTC  000016 G    $CCBNM  001520RG        $EOUT   002026RG      $RDBSZ  001524RG      $$$ARG= 000003
X$2RTT  000020 G    $CHN  = 001570RG        $ERDPB  000404RG      $RET  = 001620RG      $$$OST= 000010
X$2WSZ  000012 G    $CHNLN= 000050 G        $ERET   002027RG      $RNDCP= 001562RG      $$$T1 = 000002
X$3CT   000012 G    $CHNLS= 001634RG        $ESEG   002022RG      $RNDIA= 001755RG      $$$T2 = 000166R      003
X$3DFP  000004 G    $CHNND= 001704RG        $ESUB = 001716RG      $RNDMP= 002014RG      .FSRCB= ****** G
X$3DW   000010 G    $CIMXR= 001716RG        $EVBEG  001616RG      $RNDPA= 001556RG      .RNFSP  001512RG
X$3KM   000016 G    $CIRCT  001466RG        $EVENT  001604RG      $RNHAR= 001574RG      ...PC1= 000142R
X$3KT   000013 G    $CIRID= 001712RG        $EXSTS  001470RG      $RNHOS= 001552RG      ...PC2= 000336R
X$3LEN= 000021 G    $CIRPR= 001720RG        $FIATR  000340RG      $RNLOD= 001716RG      ...PC3= 000142R
X$3MW   000011 G    $CLASN  000162RG    003 $FILHD  001432RG      $RNSCI= 001564RG      ...TPC= 000006
X$3MXP  000006 G    $CLASS  001614RG        $FLAGS  001442RG
```

```
                                                          G 2
```

```
                                                                    F  3
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 12
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE               REFERENCES

ZF.KMX  = 000020  G         #12-1053
ZF.LLC  = 000004  G         #12-1051
ZF.LMC  = 000100  G         #12-1055
ZF.MAN  = 004000  G         #12-1061
ZF.MFL  = 000010  G         #12-1052
ZF.MTM  = 000400  G         #12-1057
ZF.MUX  = 000040  G         #12-1054
ZF.PSE  = 002000  G         #12-1060
ZF.SLI  = 010000  G         #12-1062
ZF.TIM  = 000200  G         #12-1056
ZF.X3P  = 002000  G         #12-1059
$APR    = 001572 RG         #7-230
$ASTLV    000210 RG         #6-95
$BAUD     001554 RG         #7-193
$BSZ    = 001564 RG         #7-247
$BUFF     000434 RG          6-95         6-112       #7-130
$BYT    = 001550 RG         #7-225
$BYTE     001476 RG         #7-159
$BYTMN    001506 RG         #7-163
$BYTXT    001510 RG         #7-164
$CALT   = 001576 RG         #7-249
$CCBNM    001520 RG         #7-173
$CHN    = 001570 RG         #7-238
$CHNLN  = 000050  G         #7-215
$CHNLS  = 001634 RG         #7-213        7-215
$CHNND  = 001704 RG         #7-214
$CIMXR  = 001716 RG         #7-234
$CIRCT    001466 RG         #7-155
$CIRID  = 001712 RG         #7-232
$CIRPR  = 001720 RG         #7-235
$CLASN    000162 RG         #6-95
$CLASS    001614 RG         #7-206
$CLAST    000172 RG         #6-95
$CLBUF  = 000310 RG         #6-95
$CLCSI    000212 RG         #6-95         6-95
$CLEVL    000206 RG         #6-95
$CLFDB    000266 RG         #6-95         6-95         6-95
$CLFLG    000464 RG         #6-95
$CLIOS  = 000132 RG         #6-95
$CLM      001726 RG         #7-261        7-280        7-299
$CLMCR    000000 RG         #6-95         6-95
$CLPMT    000466 RG         #6-95
$CLPRM    000472 R           6-95        #6-95         6-95
$CLQIO    000122 RG         #6-95         6-95
$CLRTN    000000 R           6-95        #6-97
$CLSBK    000176 RG          6-95        #6-95
$CLSEF    000156 RG         #6-95
$CLSIZ  = 000306 RG         #6-95
$CLSRV    000470 RG         #6-95
$CLT    = 001562 RG         #7-250
$CLTBF    000202 RG         #6-95
$CLTSZ    000204 RG         #6-95
```

```
CFEINI     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

CLOSE$           #4-79       5-125       5-136
DELET$           #4-79       5-137
ERROR$           #4-64       5-117       5-126
ERR$             #5-125      5-125      #5-136       5-136      #5-137       5-137
FINIT$           #4-79       5-106
GCL$             #4-79       5-111
LDFDB$           #5-125      5-125      #5-136       5-136      #5-137       5-137
PRINT            #4-59
```

```
                                ;+
                                ; *** - $RDATA - READ RECORD FOR INPUT FILE
                                ;
                                ; INPUT:
                                ;         $IFDB = INITIALIZED FDB
                                ;
                                ; OUTPUT:
                                ;         C-BIT = SUCCESS/FAILURE
                                ;         STRING IS TERMINATED WITH A NULL BYTE
                                ;         R0 = FDB ADDRESS
                                ;         R3 = LENGTH OF RECORD LAST READ ( FOR USE WITH TPARS )
                                ;
                                ;-

000476                          $RDATA::
000476  012700  000000G                 MOV     #$IFDB,R0       ; POINT R0 AT FDB
000502                                  GET$    ,#$BUFI         ; GET THE NEXT RECORD
000514  103404                          BCS     10$             ; IF CS, ASSUME END-OF-FILE
000516  016003  000024                  MOV     F.NRBD(R0),R3   ; GET LENGTH OF RECORD
000522  105063  000156'                 CLRB    $BUFI(R3)       ; APPEND A NULL BYTE TO THE RECORD
000526  000207                  10$:    RETURN                  ; RETURN TO CALLER
```

```
                                                 .SBTTL  PUROBJ - PURGE OBJECT
                                         ;+
                                         ; *** - PUROBJ - PURGE OBJECT
                                         ;
                                         ; INPUT:
                                         ;       $TNUM = THE OBJECT TASK NUMBER
                                         ;       $WILD = WILD CARD SPECIFICATIONS
                                         ;
                                         ; OUTPUT:
                                         ;
                                         ;-

 000226                                  PUROBJ:
 000226  012702  000000G                         MOV     #$TMLST,R2        ; GET START OF TEMPLATES
 000232  005000                                  CLR     R0                ; SUCCESS INDICATOR
                                         ;
                                         ; SEARCH FOR THE OBJECT TEMPLATE
                                         ;
 000234  010205                          10$:    MOV     R2,R5             ; COPY CURRENT TEMPLATE ADDRESS
 000236  011502                          15$:    MOV     (R5),R2           ; GET NEXT TEMPLATE
 000240  001433                                  BEQ     30$               ; IF EQ, NO MORE
 000242  122762  000000G 000000G                 CMPB    #CS.OBJ,C.STS(R2) ; IS THIS AN OBJECT TEMPLATE ?
 000250  001371                                  BNE     10$               ; IF NE, NO .. KEEP LOOKING
 000252  022767  000000G 000000G                 CMP     #WC.OBJ,$WILD     ; PURGE ALL OBJECTS ?
 000260  001404                                  BEQ     20$               ; IF EQ, YES
 000262  126762  000000G 000000G                 CMPB    $TNUM,C.PRM(R2)   ; IS THIS THE RIGHT TEMPLATE ?
 000270  001361                                  BNE     10$               ; IF NE, NO .. KEEP LOOKING
                                         ;
                                         ; PURGE OBJECT
                                         ;
 000272  004767  000746                  20$:    CALL    QREM              ; UNLINK TEMPLATE FROM LIST
 000276  012701  000013                          MOV     #OBJLEN,R1        ; GET THE LENGTH OF AN OBJECT TEMPLATE
 000302  012700  000000G                         MOV     #$XXAVL,R0        ; POINT AT END OF TASK LISTHEAD
 000306  004767  000000G                         CALL    $RLBF             ; RELEASE CORE BLOCK
 000312  162767  000000G 000000G                 SUB     #O.LENX,$BYTMN    ; ADJUST MINIMUM POOL BYTE EXTENSION
 000320  162767  000000G 000000G                 SUB     #O.LENX,$BYTXT    ; AND CURRENT BYTE EXTENSION VALUE
 000326  000743                                  BR      15$               ; LOOK FOR MORE WORK

 000330  005700                          30$:    TST     R0                ; ANY TEMPLATES FOUND ?
 000332  001004                                  BNE     40$               ; IF NE, YES .. OKAY
 000334  032767  000000G 000000G                 BIT     #WC.OBJ,$WILD     ; WAS THIS FOR KNOWN OBJECTS ?
 000342  001401                                  BEQ     101$              ; IF EQ, NO .. ERROR
 000344  000207                          40$:    RETURN
                                         ;
                                         ; ERROR CONDITIONS
                                         ;
 000346                                  101$:   ERROR$  #ERROBJ           ; NO OBJECTS FOUND
 000356  000207                                  RETURN
```

```
        .TITLE  CFEPRS - STATE TABLES TO PARSE COMMAND LINES
        .IDENT  /V05.00/
        .ENABL  LC

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1984, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - PARSE COMMAND LINES READ IN
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

DEFINE LOGGING MONITOR

```
    660 000000                          STATE$  DEFLOG
    661 000000                           TRAN$  !LOGID,OPLOG,,CO.LOG,$COMP
    662 000000                           TRAN$  !KNLOG,,,CO.KLG,$COMP
    663
    664 000000                          STATE$  OPLOG,&P$DELO
    665 000000                           TRAN$  !CPSTA,OPLOG1,,OM.STA,$OPRND
    666 000000                           TRAN$  !OPEVE,OPLOG2,,OM.EVE,$OPRND
    667
    668 000000                          STATE$  OPLOG1
    669 000000                           TRAN$  !OPEVE,$EXIT,,OM.EVE,$OPRND
    670 000000                           TRAN$  $LAMDA,$EXIT
    671
    672 000000                          STATE$  OPLOG2
    673 000000                           TRAN$  !OPSTA,$EXIT,,OM.EVE,$OPRND
    674 000000                           TRAN$  $LAMDA,$EXIT
    675
    676 000000                          STATE$  OPSTA
    677 000000                           TRAN$  &K$STA,,,OM.STA,$OPRND
    678
    679 000000                          STATE$  ,&P$STA
    680 000000                           TRAN$  &K$ON,$EXIT,,SW.LON,$SWICH
    681 000000                           TRAN$  &K$OFF,$EXIT
    682
    683 000000                          STATE$  OPEVE
    684 000000                           TRAN$  &K$KNO,OPEVE1
    685 000000                           TRAN$  &K$EVE
    686
    687 000000                          STATE$  ,&P$EVE
    688 000000                           TRAN$  !EVEID,$EXIT
    689
    690 000000                          STATE$  OPEVE1
    691 000000                           TRAN$  &K$EVE,$EXIT,,OM.KEV,$OPRND
    692                                 .SBTTL  SUB-EXEPRESSIONS FOR EVENT ID
    693
    694                         ;
    695                         ; EVENT IDENTIFICATION
    696                         ;
    697 000000                          STATE$  EVEID
    698 000000                           TRAN$  $LAMDA,,$BL.MAN
    699 000000                          STATE$
    700 000000                           TRAN$  $DNUMB,,CKEVEC
    701 000000                          STATE$
    702 000000                           TRAN$  <'.>
    703 000000                          STATE$
    704 000000                           TRAN$  <'*>,EVEID2,,WC.EVE,$WILD
    705 000000                           TRAN$  $LAMDA
    706 000000                          STATE$
    707 000000                           TRAN$  !EVELIS
    708 000000                          STATE$  EVEID1
    709 000000                           TRAN$  $EOS,EVEID2
    710 000000                           TRAN$  $BLANK,EVEID2
    711 000000                           TRAN$  <',>
    712 000000                          STATE$
    713 000000                           TRAN$  !EVELIS,EVEID1
    714 000000                          STATE$  EVEID2
    715 000000                           TRAN$  $LAMDA,$EXIT,$BL.AUT
    716                         ;
```

```
   1322 000000                                  STATE$  DEFLIN
   1323 000000                                   TRAN$  !LINID,LINE,,CO.LIN,$COMP
   1324 000000                                   TRAN$  !KNLIN,,,CO.KNL,$COMP
   1325
   1326 000000                                  STATE$  LINE,&P$DELI
   1327 000000                                   TRAN$  !LINOPT
   1328
   1329 000000                                  STATE$  LINO
   1330 000000                                   TRAN$  !LINOPT,LINO
   1331 000000                                   TRAN$  $LAMDA,$EXIT
   1332                                 ;
   1333                                 ; LINE OPTIONS
   1334                                 ;
   1335 000000                                  STATE$  LINOPT
   1336 000000                                   TRAN$  !LSTATE,$EXIT,, OL.STA, $OPRND
   1337 000000                                   TRAN$  !BAUD,  $EXIT,, OL.BAU, $OPRND
   1338 000000                                   TRAN$  !CSR,   $EXIT,, OL.CSR, $OPRND
   1339 000000                                   TRAN$  !UNCSR, $EXIT,, OL.SEC, $OPRND
   1340 000000                                   TRAN$  !VEC,   $EXIT,, OL.VEC, $OPRND
   1341 000000                                   TRAN$  !PRI,   $EXIT,, OL.PRI, $OPRND
   1342 000000                                   TRAN$  !PLX,   $EXIT,, OL.PLX, $OPRND
   1343 000000                                   TRAN$  !URM,   $EXIT,, OL.URM, $OPRND
   1344 000000                                   TRAN$  !DPR,   $EXIT,, OL.DPR, $OPRND
   1345 000000                                   TRAN$  !LCT,   $EXIT,, OL.CTM, $OPRND
   1346 000000                                   TRAN$  !HBT,   $EXIT,, OL.HBT, $OPRND
   1347 000000                                   TRAN$  !MBL,   $EXIT,, OL.BSZ, $OPRND
   1348 000000                                   TRAN$  !MRT,   $EXIT,, OL.RTC, $OPRND
   1349 000000                                   TRAN$  !MWIL,  $EXIT,, OL.WSZ, $OPRND
   1350 000000                                   TRAN$  !RTT,   $EXIT,, OL.RTT, $OPRND
   1351                                 ;
   1352                                 ; LOOK FOR LINE  STATE
   1353                                 ;
   1354 000000                                  STATE$  LSTATE
   1355 000000                                   TRAN$  &K$STA
   1356
   1357 000000                                  STATE$  ,&P$LIST
   1358 000000                                   TRAN$  &K$ON,  $EXIT,, SW.LOA!SW.ENA,  $SWICH
   1359 000000                                   TRAN$  &K$OFF, $EXIT,, SW.LOA,         $SWICH
   1360 000000                                   TRAN$  &K$CLE, $EXIT
   1361                                 ;
   1362                                 ; LOOK FOR VECTOR ADDRESS
   1363                                 ;
   1364 000000                                  STATE$  VEC
   1365 000000                                   TRAN$  &K$VEC
   1366
   1367 000000                                  STATE$  ,&P$VEC
   1368 000000                                   TRAN$  $NUMBR, $EXIT,  .LIVEC
   1369                                 ;
   1370                                 ; LOOK FOR PRIORITY
   1371                                 ;
   1372 000000                                  STATE$  PRI
   1373 000000                                   TRAN$  &K$PRI
   1374
   1375 000000                                  STATE$  ,&P$PRI
   1376 000000                                   TRAN$  $NUMBR, $EXIT,  .LIPRI
   1377                                 ;
   1378                                 ; LOOK FOR DUPLEX
```

```
                                        .SBTTL  DEFINE SYSTEM COMMAND
                                ;
                                ; DEFINE SYSTEM COMMAND
                                ;
 000000                                 STATE$  DEFSYS
 000000                                  TRAN$  &K$SYS,,,               CO.SYS, $COMP

 000000                                 STATE$  ,&P$SYS
 000000                                  TRAN$  !SYSOPT

 000000                                 STATE$  SYSO
 000000                                  TRAN$  $EOS,    $EXIT
 000000                                  TRAN$  !SYSOPT,SYSO
                                ;
                                ; DEFINE SYSTEM OPTIONS
                                ;
 000000                                 STATE$  SYSOPT
 000000                                  TRAN$  !POOL,   $EXIT
 000000                                  TRAN$  !LOC,    $EXIT,,        OS.LOC, $OPRND
 000000                                  TRAN$  !CCBNM,  $EXIT,,        OS.CCB, $OPRND
 000000                                  TRAN$  !RDBNM,  $EXIT,,        OS.RBN, $OPRND
 000000                                  TRAN$  !RDBSZ,  $EXIT,,        OS.RBS, $OPRND
 000000                                  TRAN$  !SDBNM,  $EXIT,,        OS.SBN, $OPRND
 000000                                  TRAN$  !THRESH,$EXIT,,         OS.THR, $OPRND

                                ;
                                ; DEFINE SYSTEM POOL
                                ;
 000000                                 STATE$  POOL
 000000                                  TRAN$  &K$POL

 000000                                 STATE$  ,&P$POPT
 000000                                  TRAN$  !PNAM,   $EXIT,,        OS.NAM, $OPRND
 000000                                  TRAN$  !PAR,    $EXIT,,        OS.PAR, $OPRND
 000000                                  TRAN$  !BYT,    $EXIT,,        OS.BYT, $OPRND
                                ;
                                ; POOL NAME
                                ;
 000000                                 STATE$  PNAM
 000000                                  TRAN$  &K$NAM

 000000                                 STATE$  ,&P$PNAM
 000000                                  TRAN$  $RAD50, $EXIT,   .SYPOL
                                ;
                                ; PARTITION NAME
                                ;
 000000                                 STATE$  PAR
 000000                                  TRAN$  &K$PAR

 000000                                 STATE$  ,&P$PAR
 000000                                  TRAN$  $RAD50, $EXIT,   .SYPAR
                                ;
                                ; BYTE EXTENSION
                                ;
 000000                                 STATE$  BYT
 000000                                  TRAN$  &K$BYT

```

```
                                        .SBTTL  EXIT STATE

                                ;
                                ; EXIT STATE
                                ;
000000                                  STATE$  EXIT
000000                                   TRAN$  $EOS,   $EXIT

000000                                  STATE$


        000001                          .END
        .TITLE  CFEROT - ROOT SEGMENT FOR CFE
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - ROOT SEGMENT
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
;
; IDENT HISTORY:
;
; 1.00  10-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
 000140  060104                           ADD     R1,R4           ;ADD IN BLOCK SIZE
 000142  001404                           BEQ     3$              ;EQ - NEW BLOCK ENDS AT 32K - DO NOT MERGE
 000144  020304                           CMP     R3,R4           ;COMPARE ADDRESSES
 000146  001002                           BNE     3$              ;IF NE DO NOT MERGE BLOCKS
 000150  012312                           MOV     (R3)+,(R2)      ;SET NEW FORWARD LINK
 000152  061301                           ADD     (R3),R1         ;ADJUST SIZE
                                  ;
                                  ; SEE IF BLOCKS CAN BE MERGED IN OLD + NEW ORDER.
                                  ;
 000154  010210                   3$:     MOV     R2,(R0)         ;ASSUME NO AGGLOMERATION
 000156  010004                           MOV     R0,R4           ;CALCULATE ADDRESS OF NEXT BLOCK
 000160  066004  000002                   ADD     2(R0),R4        ;ADD IN BLOCK SIZE
 000164  020204                           CMP     R2,R4           ;COMPARE ADDRESSES
 000166  001004                           BNE     4$              ;IF NE DO NOT MERGE
 000170  011210                           MOV     (R2),(R0)       ;SET NEW FORWARD LINK
 000172  066001  000002                   ADD     2(R0),R1        ;ADJUST SIZE
 000176  010002                           MOV     R0,R2           ;SET NEW BASE ADDRESS OF BLOCK
 000200  010162  000002           4$:     MOV     R1,2(R2)        ;SET SIZE OF BLOCK
 000204  000207                   5$:     RETURN                  ;

                                  ;+
                                  ; **-.RQCB-REQUEST CORE BLOCK
                                  ;
                                  ; THIS ROUTINE IS CALLED TO REQUEST A CORE BLOCK FROM THE FREE LIST.
                                  ; THE SELECTION IS MADE EITHER FIRST FIT OR BEST FIT ACCORDING TO THE
                                  ; THE INPUT PARAMETERS.
                                  ;
                                  ; INPUTS:
                                  ;
                                  ;       R0=ADDRESS OF FREE BLOCK LIST HEAD.
                                  ;       R1=SIZE OF BLOCK REQUESTED IN BYTES.
                                  ;               IF R1 IS GE 0, THEN BEST FIT IS USED.
                                  ;               IF R1 IS LT 0, THEN FIRST FIT IS USED.
                                  ;
                                  ;       IN EITHER CASE THE MAGNITUDE OF R1 IS THE SIZE OF THE BLOCK.
                                  ;
                                  ; OUTPUTS:
                                  ;
                                  ;       C=1 IF REQUEST CANNOT BE FULFILLED.
                                  ;       C=0 IF REQUEST IS SATISFIED.
                                  ;               R0=ADDRESS OF REQUESTED CORE BLOCK.
                                  ;
                                  ;-

 000206  004567  000000G          .RQCB:: JSR     R5,$SAVRG       ;SAVE NONVOLITILE REGISTERS
 000212  005004                           CLR     R4              ;CLEAR PREVIOUS BLOCK ADDRESS
 000214  005005                           CLR     R5              ;CLEAR FOUND BLOCK ADDRESS
 000216  010102                           MOV     R1,R2           ;SET TYPE OF FIT FLAG
 000220  100001                           BPL     1$              ;IF PL BEST FIT
 000222  005401                           NEG     R1              ;MAKE SIZE POSITIVE
 000224  062701  000003           1$:     ADD     #3,R1           ;ROUND TO NEXT 2 WORD BOUNDRY
 000230  042701  000003                   BIC     #3,R1           ;CLEAR EXCESS BITS
 000234  001442                           BEQ     9$              ;IF EQ VERGESSEN!!!
 000236  011003                   2$:     MOV     (R0),R3         ;GET ADDR OF NEXT
 000240  001420                           BEQ     5$              ;IF EQ END OF LIST
 000242  020163  000002                   CMP     R1,2(R3)        ;BLOCK BIG ENOUGH?
 000246  101013                           BHI     4$              ;IF HI NO
 000250  001422                           BEQ     6$              ;IF EQ CORRECT SIZE--USE IT
```

```
Symbol table

CM.EXI= ****** GX       $DNUMB= 000014          $TMCHN= ****** GX       $TMLST= ****** GX       $TMSTA= ****** GX
ERRALL= ****** GX       $EOS  = 000012          $TMCNT= ****** GX       $TMNOD= ****** GX       $TMSTB  000002RG
ERRSYN= ****** GX       $ERROR  000000RG        $TMCUG= ****** GX       $TMOBJ= ****** GX       $TMSVC= ****** GX
RBRKT = 000076          $EXIT = 000000          $TMDDM= ****** GX       $TMPAR= ****** GX       $TMUNT= ****** GX
SEMI  = 000073          $FAIL = 177777          $TMDEC= ****** GX       $TMPSN= ****** GX       $TMX2P= ****** GX
TMPKTB  000000RG    003 $LAMDA= 000000          $TMDLC= ****** GX       $TMPVC= ****** GX       $TMX29= ****** GX
TMPSTB  000000RG    002 $NUMBR= 000002          $TMDSA= ****** GX       $TMRDT= ****** GX       $TMX3P= ****** GX
$ALPHA= 000022          $QINS = ****** GX       $TMDSC= ****** GX       $TMREM= ****** GX       $T1ALC  000170RG
$ANY  = 000020          $RAD50= 000016          $TMDST= ****** GX       $TMROU= ****** GX       $XXAVL= ****** GX
$BLANK= 000006          $RDATA= ****** GX       $TMDTE= ****** GX       $TMSAC= ****** GX       $$$FLG= 177777
$BUFI = ****** GX       $RQBF = ****** GX       $TMEVT= ****** GX       $TMSAD= ****** GX       $$$KEY= 000044
$CFERR= ****** GX       $STRNG= 000004          $TMFEA= ****** GX       $TMSER= ****** GX       $$$STA= 000000
$CFER1= ****** GX       $SUBXP= 000010          $TMFIL= ****** GX       $TMSLT= ****** GX       $$$TMP= 000374R     004
$CMAND= ****** GX       $TALOC  000146RG        $TMLLC= ****** GX       $TMSNA= ****** GX       .EXIT = ****** GX
$DIGIT= 000024          $TMBUF= ****** GX       $TMLOG= ****** GX       $TMSPU= ****** GX       .TPARS= ****** GX


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000234    001   (RW,I,LCL,REL,CON)
$STATE  000344    002   (RW,D,LCL,REL,CON)
$KTAB   000112    003   (RW,D,LCL,REL,CON)
$KSTR   000403    004   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10646 Words  ( 42 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:35.31
SY:CFESTB.V2,[132,134]CFESTB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFESTB
```

```
                                                G 14
CFESTB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14     PAGE 1
```

```
                                           .SBTTL  $C5TA - CONVERT EXTENDED RAD50 TO ASCII ("*")
                                           .SBTTL  $C5TA1- CONVERT EXTENDED RAD50 TO ASCII ("-")
                                   ;+
                                   ; *** - $C5TA - CONVERT EXTENDED RAD50 TO ASCII (WITH TRAILING SPACES DELETED)
                                   ; *** - $C5TA1- CONVERT EXTENDED RAD50 TO ASCII ("-")
                                   ;
                                   ; INPUT:
                                   ;          R0 = ADDRESS OF OUTPUT BUFFER
                                   ;          R1 = RAD50 CHARACTER TO CONVERT
                                   ;
                                   ; OUTPUTS:
                                   ;          R0 = NEXT LOCATION IN BUFFER AFTER CONVERTED STRING
                                   ;
                                   ;          EXTENDED RAD50 INCLUDES THE "*" (OR "-") CHARACTER.
                                   ;
                                   ;-

000232                             EXTCHR: .BLKW   1                       ; EXTENDED RAD50 CHAR

                                           .ENABL  LSB

000234                             $C5TA1::
000234  012767  000033  177770             MOV     #<'--22>,EXTCHR         ; EXTEND BY HYPHEN
000242  000403                             BR      5$                      ; JOIN COMMON CODE
000244                             $C5TA::
000244  012767  000030  177760             MOV     #<'*-22>,EXTCHR         ; EXTEND BY ASTERISK
000252  010002                     5$:     MOV     R0,R2                   ; SAVE BYTE ADDRESS
000254  004767  000036                     CALL    CVTC                    ; CONVERT FIRST CHAR
000260  010046                             MOV     R0,-(SP)                ; SAVE IT
000262  004767  000030                     CALL    CVTC                    ; CONVERT SECOND CHAR
000266  010046                             MOV     R0,-(SP)                ; SAVE IT
000270  004767  000022                     CALL    CVTC                    ; CONVERT THIRD CHAR
000274  110022                             MOVB    R0,(R2)+                ; STORE IT
000276  112622                             MOVB    (SP)+,(R2)+             ; STORE SECOND CHAR
000300  112622                             MOVB    (SP)+,(R2)+             ; STORE THIRD CHAR
000302  010200                             MOV     R2,R0                   ; SAVE TERMINAL ADDRESS
000304  122740  000040             10$:    CMPB    #' ,-(R0)               ; BACK UP OVER SPACES
000310  001775                             BEQ     10$                     ; ...
000312  005200                             INC     R0                      ; WENT A LITTLE TOO FAR !
000314  000207                             RETURN

                                           .DSABL  LSB

                                   ;
                                   ; CONVERT RAD50 CHARACTER
                                   ;
000316  010100                     CVTC:   MOV     R1,R0                   ; COPY DIVIDEND TO R0
000320  012701  000050                     MOV     #50,R1                  ; SET DIVISOR
000324  004767  000000G                    CALL    $DIV                    ; DIVIDE EM UP
000330  010146                             MOV     R1,-(SP)                ; SAVE REMAINDER
000332  010001                             MOV     R0,R1                   ; COPY QUOTIENT TO R1
000334  012600                             MOV     (SP)+,R0                ; RETRIEVE REMAINDER
000336  001417                             BEQ     30$                     ; IF EQ, BLANK
000340  020027  000033                     CMP     R0,#33                  ; TEST MIDDLE
000344  002412                             BLT     20$                     ; IF LT, ALPHA
000346  001407                             BEQ     10$                     ; IF EQ, DOLLAR
000350  020027  000035                     CMP     R0,#35                  ; TEST FOR ASTERISK
000354  001002                             BNE     5$                      ; IF NE, NOT ASTERISK
000356  016700  177650                     MOV     EXTCHR,R0               ; OUTPUT ASTERISK (OR HYPHEN)
```

```
                                                                           F 16
CFESUB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE              REFERENCES

CD.LEN  = 000022           #4-64
CNTRL     000404 R         #12-354      *12-363     *12-366      12-379
CS.DDM  = ******  GX         13-458
CS.DEC  = ******  GX         16-591
CS.DLC  = ******  GX         13-456
CS.LLC  = ******  GX         13-454
CS.MOD  = ******  GX         16-623
CS.PAR  = ******  GX         16-587
CS.ROU  = ******  GX         16-595
CS.SLT  = ******  GX         14-490
CS.STA  = ******  GX         15-548
CVTBL     000414 R         #12-357       12-377      12-423
CVTC      000316 R           11-301      11-303      11-305     #11-319
C$LEN     000016              5-97
C$LENX  = 000020  G         #5-97
C.CNT     000010            #4-75        14-508
C.DELF    000016            #4-64
C.DELW    000017            #4-64
C.FLG     000006            #4-74        14-499      14-503      14-516
C.INAC    000014            #4-64
C.INCT    000012            #4-64
C.IPL     000020            #4-64
C.LKS     000010            #4-64
C.LST     000021            #4-64
C.NAM     000004            #4-73        14-496
C.NDC     000006            #4-64        16-604
C.OUTT    000013            #4-64
C.PRM   = ******  GX         13-462      13-465      15-553      15-555      16-597      16-615     *16-622
C.RETF    000015            #4-64
C.SEG     000004            #4-64
C.STS   = ******  GX         13-454      13-456      13-458      14-490      15-548      16-587      16-591      16-595     *16-623
C.UNT     000011            #4-76        14-514
D$FLEN    000032              5-101
D$FLNX  = 000034  G         #5-101
EXTCHR    000232 R          #11-291     *11-296     *11-299      11-331
EX$SUC  = ******  GX          8-196
E$NBR     000014            #4-63
E$NBS     000020            #4-63
E$NCR     000034            #4-63
E$NCS     000036            #4-63
E$NIC     000044            #4-63
E$NLEN    000050            #4-63        16-605
E$NLLA    000012            #4-63
E$NLNK    000000            #4-63
E$NML     000040            #4-63
E$NMR     000024            #4-63
E$NMS     000030            #4-63
E$NNOD    000002            #4-63
E$NRT     000042            #4-63
E$NRTP    000005            #4-63
E$NSEG    000010            #4-63
E$NTIM    000046            #4-63
```

```
                                                                           G 16
CFESUB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 2
```

```
   748          000001                            WC.NAM == 1             ; =1 IF WILD CARD NAMES
   749          000002                            WC.CNT == 2             ; =1 IF WILD CARD CONTROLLER
   750          000004                            WC.UNT == 4             ; =1 IF WILD CARD UNIT
   751          000010                            WC.TRI == 10            ; =1 IF WILD CARD TRIBUTARY
   752          000017                              WC.LIN == 17          ; WILD CARD LINES
   753          000017                              WC.CIR == 17          ; WILD CARD CIRCUITS
   754          000001                              WC.PRO == 1           ; WILD CARD PROCESSES
   755          000001                              WC.OBJ == 1           ; WILD CARD OBJECTS
   756          000001                              WC.NOD == 1           ; WILD CARD NODES
   757          000001                              WC.EVE == 1           ; WILD CARD EVENTS
   758          000002                              WC.LOG == 2           ; WILD CARD LOGGING
   759                                       ;
   760                                       ;*****
   761                                       ; OPERAND BIT DEFINITIONS
   762                                       ;
   763                                       ; LINE OPTIONS
   764                                       ;
   765          000001                            OL.CTM == 1             ; =1 IF COUNTER TIMER IS PRESENT
   766          000002                            OL.STA == 2             ; =1 IF LOAD REQUEST
   767
   768          000004                            OL.PRI == 4             ; =1 IF PRIORITY PRESENT
   769          000010                            OL.PLX == 10            ; =1 IF DUPLEX IS PRESENT
   770          000020                            OL.VEC == 20            ; =1 IF VECTOR IS PRESENT
   771          000040                            OL.BAU == 40            ; =1 IF BAUD RATE IS PRESENT
   772          000100                            OL.SEC == 100           ; =1 IF SECONDARY CSR IS PRESENT
   773          000200                            OL.CSR == 200           ; =1 IF CSR IS PRESENT
   774          000400                            OL.URM == 400           ; =1 IF UNIBUS RUN MASK PRESENT
   775          001000                            OL.DPR == 1000          ; =1 IF DEAD POLLING RATIO PRESENT
   776          002000                            OL.HBT == 2000          ; =1 IF HOLDBACK TIMER IS PRESENT
   777          004000                            OL.RTC == 4000          ; =1 IF RETRANSMIT COUNT IS PRESENT
   778          010000                            OL.WSZ == 10000         ; =1 IF WINDOW SIZE IS PRESENT
   779          020000                            OL.BSZ == 20000         ; =1 IF BLOCK SIZE IS PRESENT
   780          040000                            OL.RTT == 40000         ; =1 IF RETRANSMIT TIMER IS PRESENT
   781          000003                              OL.SLT == OL.STA!OL.CTM ; SLT PARAMETERS
   782          076000                              OL.X25 == OL.HBT!OL.RTC!OL.BSZ!OL.WSZ!OL.RTT ; X.25 LEVEL 2 PARAMETERS
   783          000624                              OL.CNT == OL.PRI!OL.VEC!OL.CSR!OL.URM ; CONTROLLER PARAMETERS
   784          000000                              OL.STN == 0                           ; STATION PARAMETERS
   785          001150                              OL.UNT == OL.PLX!OL.SEC!OL.BAU!OL.DPR!OL.STN ; UNITS/STATIONS
   786          000320                              OL.NWC == OL.VEC!OL.SEC!OL.CSR        ; NO WILDCARDS ON THESE
   787                                       ;
   788                                       ; CIRCUIT OPTIONS
   789                                       ;
   790
   791          000001                            OC.CTM  == 1            ; COUNTER TIMER         (COMMON WITH LINE)
   792          000002                            OC.STA  == 2            ; STATE                 (COMMON WITH LINE)
   793
   794          000004                            OC.CST  == 4            ; COST
   795          000010                            OC.MWI  == 10           ; MAXIMUM WINDOWS       (PSI)
   796
   797          000020                            OC.DTE  == 20           ; DTE ADDRESS           (PSI)
   798          000040                            OC.CHN  == 40           ; CHANNEL               (PSI)
   799          000100                            OC.MBL  == 100          ; MAXIMUM BLOCKS        (PSI)
   800          000200                            OC.TRI  == 200          ; TRIBUTARY ADDRESS
   801
   802          000400                            OC.RTM  == 400          ; RECALL TIMER          (PSI)
   803          001000                            OC.MRC  == 1000         ; MAXIMUM RECALLS       (PSI)
   804
```

```
CFEIMP - SYMBOL DEFINITIONS AND MACRO V05.03b Saturday 29-Jun-85 04:55  Page 13-4
Symbol table

. ABS.  000062    000   (RW,I,GBL,ABS,OVR)
        002660    001   (RW,I,LCL,REL,CON)
$$FSR1  003060    002   (RW,D,GBL,REL,OVR)
$$$GCL  000502    003   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13449 Words  ( 53 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:40.16
SY:CFEIMP.V2,[132,134]CFEIMP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEIMP
```

G 3

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 13
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE             REFERENCES

$CLV       001756 RG       #7-263
$CLWEF     000152 RG       #6-95
$CMAND     001446 RG       #7-142
$CMDPT   = 001450 RG       #7-228
$CNFIG     001440 RG       #7-140
$CNTNM     001534 RG       #7-183
$COMP      001450 RG       #7-143      7-228
$COPES     001574 RG       #7-200      7-222       7-237
$COST      001710 RG       #7-217
$CSR       001556 RG       #7-194      7-221
$CT      = 001574 RG       #7-237
$CUG     = 001576 RG       #7-248
$DEFNB     000302 RG        6-106      6-113      #6-115
$DFD     = 001602 RG       #7-251
$DFW     = 001540 RG       #7-252
$DPR     = 001570 RG       #7-229      7-238       7-256
$DST       001716 RG       #7-233      7-234       7-235       7-236
$DTE       001622 RG       #7-209
$DVNAM   = 001532 RG       #7-182
$EACOS     002012 RG       #7-267
$EAHOP     002014 RG       #7-268
$EAMAX     002010 RG       #7-266
$EBEA      002020 RG       #7-270
$EBRT      002016 RG       #7-269
$EINA      002024 RG       #7-272
$EINC      002025 RG       #7-273
$EOUT      002026 RG       #7-274
$ERDPB     000404 RG       #7-128
$ERET      002027 RG       #7-275
$ESEG      002022 RG       #7-271
$ESUB    = 001716 RG       #7-236
$EVBEG     001616 RG       #7-207      7-245
$EVENT     001604 RG       #7-205
$EXSTS     001470 RG       #7-156
$FIATR     000340 RG       #6-117
$FILHD     001432 RG       #7-136
$FLAGS     001442 RG       #7-141
$HBT     = 001602 RG       #7-244
$HIVBN     001472 RG       #7-157
$HOST    = 001560 RG       #7-231
$HTM     = 001620 RG       #7-239
$ID        001622 RG       #7-210      7-212       7-213       7-290
$IFDB      000142 RG       #6-110
$INFO      001464 RG       #7-148
$LINKS     001562 RG       #7-196      7- 27       7-250
$LOVBN     001474 RG       #7-158
$LSNAD     001436 RG       #7-138
$LSTMP     001434 RG       #7-137
$MAXC    = 001550 RG       #7-219
$MAXH    = 001552 RG       #7-220
$MBL     = 001564 RG       #7-240
$MCI     = 001542 RG       #7-257
```

H 3

G 4

```
**FILE**ID**CFELIS




  CCCCCCCC   FFFFFFFFFF    EEEEEEEEEE   LL             IIIIII        SSSSSSSS
  CCCCCCCC   FFFFFFFFFF    EEEEEEEEEE   LL             IIIIII        SSSSSSSS
CC           FF            EE           LL               II        SS
CC           FF            EE           LL               II        SS
CC           FF            EE           LL               II        SS
CC           FF            EE           LL               II        SS
CC           FFFFFFFF      EEEEEEEE     LL               II          SSSSSS
CC           FFFFFFFF      EEEEEEEE     LL               II          SSSSSS
CC           FF            EE           LL               II                SS
CC           FF            EE           LL               II                SS
CC           FF            EE           LL               II                SS       ....
CC           FF            EE           LL               II                SS       ....
  CCCCCCCC   FF            EEEEEEEEEE   LLLLLLLLLL     IIIIII      SSSSSSSS         ....
  CCCCCCCC   FF            EEEEEEEEEE   LLLLLLLLLL     IIIIII      SSSSSSSS         ....




LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

H 4

```
                                        ;+
                                        ; *** - $OPEN - OPEN THE OUTPUT FILE
                                        ;
                                        ; INPUTS:
                                        ;         R1 = ADDRESS OF FILE-SPEC
                                        ;         R2 = LENGTH OF FILE-SPEC
                                        ;
                                        ; OUTPUTS:
                                        ;         C-BIT = SUCCESS/FAILURE
                                        ;         R0 = FDB ADDRESS
                                        ;         R1,R2 = PRESERVED
                                        ;         IF SUCCESS: FILE IS OPENED FOR WRITE
                                        ;
                                        ;
                                        ;-

000530                                  $OPEN::
000530  010146                                  MOV     R1,-(SP)        ; SAVE REGISTER
000532  012700  000300'                         MOV     #CSBLK,R0       ; POINT R0 AT CSI CONTROL BLOCK
000536  010260  000002                          MOV     R2,C.CMLD(R0)   ; GET LENGTH OF FILE-SPEC STRING
000542  010160  000004                          MOV     R1,C.CMLD+2(R0) ; AND ADDRESS OF FILE-SPEC
000546                                          CSI$1                   ; PARSE FILESTRING
000552  103404                                  BCS     10$             ; IF CS, ERROR
000554                                          CSI$2   ,OUTPUT         ; GET OUTPUT FILE SPEC
000564  112767  000000G 000052G 10$:            MOVB    #IE.BNM,F.ERR+$OFDB ; ASSUME BAD FILENAME
000572  103431                                  BCS     40$             ; IF CS, RETURN WITH ERROR CODE
                                        ;
                                        ; IGNORE INPUT VERSON NUMBER AND OPEN NEW OUTPUT VERSION
                                        ;
000574  016700  177520                          MOV     CSBLK+C.FILD+2,R0; POINT TO FILENAME
000600  016701  177512                          MOV     CSBLK+C.FILD+0,R1 ; GET LENGTH
000604  010146                                  MOV     R1,-(SP)        ; SAVE IT
000606  060100                                  ADD     R1,R0           ; POINT TO CHARACTER AFTER LAST
000610  005301                          20$:    DEC     R1              ; REDUCE LENGTH
000612  002405                                  BLT     30$             ; IF LT, NO VERSION - DONE
000614  124027  000073                          CMPB    -(R0),#<';>     ; CHECK FOR SEMI
000620  001373                                  BNE     20$             ; IF NE, KEEP ON SCANNIN'
000622  010167  177470                          MOV     R1,CSBLK+C.FILD+0 ; SHORTEN TO SKIP VERSION
000626  012700  000000G                 30$:    MOV     #$OFDB,R0       ; POINT AT FDB
000632                                          FDOP$R  ,,#CSBLK+C.DSDS ; POINT AT PARSED FILE-SPEC
000640                                          OPEN$W                  ; OPEN OUTPUT FILE
000652  012667  177440                          MOV     (SP)+,CSBLK+C.FILD+0 ;  RESTORE VERSION
000656  012601                          40$:    MOV     (SP)+,R1        ; RESTORE REGISTER
000660  000207                                  RETURN                  ; RETURN TO CALLER

```

```
                                                .SBTTL  PURLOG - PURGE LOGGING

                                        ;+
                                        ;
                                        ; $PURLOG - PURGE LOGGING
                                        ;
                                        ; INPUTS:
                                        ;       $WILD - WILD CARD SPECIFICATIONS
                                        ;       $EVENT - EVENT MASKS
                                        ;       $CLASS - EVENT CLASS
                                        ;       $OPRND - LOGGING OPTIONS
                                        ;
                                        ; OUTPUT:
                                        ;       TEMPLATE IS UPDATED
                                        ;
                                        ;-
000360  032767  000000G 000000G PURLOG: BIT     #OM.KEV,$OPRND          ; KNOWN EVENTS SPECIFIED?
000366  001425                          BEQ     20$                     ; BR IF NO
000370  052767  000000G 000000G         BIS     #WC.EVE,$WILD           ; WILDCARD EVENTS IF KNOWN EVENTS
000376  012767  000000G 177606          MOV     #$KNEVT,TEMP            ; POINT TO KNOWN EVENTS TABLE
000404  016704  177602          10$:    MOV     TEMP,R4                 ; RETRIEVE KNOWN EVENTS TABLE POINTER
000410  012467  000000G                 MOV     (R4)+,$CLASS            ; GET EVENT CLASS
000414  100546                          BMI     80$                     ; BR IF END OF TABLE
000416  012467  000000G                 MOV     (R4)+,$EVENT+0          ; GET FILTER MASKS
000422  012467  000002G                 MOV     (R4)+,$EVENT+2          ; ...
000426  012467  000004G                 MOV     (R4)+,$EVENT+4          ; ...
000432  012467  000006G                 MOV     (R4)+,$EVENT+6          ; ...
000436  010467  177550                  MOV     R4,TEMP                 ; SAVE UPDATED KNOWN EVENTS POINTER

000442  032767  000000G 000000G 20$:    BIT     #WC.LOG,$WILD           ; KNOWN LOGGING SPECIFIED?
000450  001017                          BNE     30$                     ; BR IF YES
000452  012704  000001                  MOV     #FF.CON,R4              ; ASSUME LOGGING CONSOLE
000456  032767  000000G 000000G         BIT     #OM.CON,$OPRND          ; LOGGING CONSOLE SPECIFIED?
000464  001015                          BNE     45$                     ; BR IF YES
000466  012704  000002                  MOV     #FF.FIL,R4              ; ASSUME LOGGING FILE
000472  032767  000000G 000000G         BIT     #OM.FIL,$OPRND          ; LOGGING FILE SPECIFIED?
000500  001007                          BNE     45$                     ; BR IF YES
000502  012704  000004                  MOV     #FF.MON,R4              ; ELSE MUST BE LOGGING MONITOR
000506  000404                          BR      45$                     ; CONTINUE

000510  012703  000202'         30$:    MOV     #SNKTBL,R3              ; POINT TO SINK TYPE TABLE
000514  012304                  40$:    MOV     (R3)+,R4                ; GET SINK TYPE
000516  001501                          BEQ     75$                     ; BR IF END OF LIST

000520  012702  000000G         45$:    MOV     #$TMLST,R2              ; GET START OF TEMPLATES
000524  010205                  50$:    MOV     R2,R5                   ; COPY CURRENT TEMPLATE ADDRESS
000526  011502                          MOV     (R5),R2                 ; GET NEXT TEMPLATE
000530  001470                          BEQ     70$                     ; BR IF END OF LIST
000532  122762  000000G 000000G         CMPB    #CS.EVT,C.STS(R2)       ; IS THIS AN EVENT TEMPLATE?
000540  001371                          BNE     50$                     ; BR IF NO
000542  026762  000000G 000004          CMP     $CLASS,C.CLS(R2)        ; IS THIS THE CORRECT TEMPLATE?
000550  001365                          BNE     50$                     ; BR IF NO
000552  020462  000016                  CMP     R4,C.SNK(R2)            ; IS THIS THE CORRECT TEMPLATE?
000556  001362                          BNE     50$                     ; BR IF NO
000560  010201                          MOV     R2,R1                   ; POINT TO EVENT TEMPLATE
000562  062701  000006                  ADD     #C.EVT,R1               ; POINT TO EVENT MASKS
000566  032767  000000G 000000G         BIT     #WC.EVE,$WILD           ; ALL EVENTS?
```

```
     54                                                   .SBTTL  MACROS AND STORAGE
     55
     56                                          ;****
     57                                          ; LIBRARY MACRO CALLS
     58                                          ;****
     59
     60                                                   .MCALL  ISTAT$,STATE$,TRAN$,GCL$,KEYWD$
     61
     62                                          ;
     63                                          ; SIGNAL PROMPTING SUPPORT
     64                                          ;
     65              000001                               $GPRM = 1
     66                                          ;
     67                                          ; ONLY DEFINE OFFSETS FOR PROMPT STRINGS
     68                                          ;
     69              000001                               $GOFF = 1
     70                                          ;
     71                                          ; MAKE TPARS TABLES READ ONLY
     72                                          ;
     73              000001                               $RONLY = 1
     74
     75
     76              000042                      QUOTE = '"                           ; QUOTED STRING DELIMETER
     77              000076                      RTBRK = '>                           ; RIGHT BRACKET
     78                                                   .ENABL  LC
     79                                                   .SBTTL  KEYWORDS
     80                                          ;****
     81                                          ; TPARS STATE TABLES
     82                                          ;****
     83
     84 000000                                            ISTAT$  PRSSTB,PRSKTB
     85                                          ;
     86                                          ; DEFINE KEYWORDS
     87                                          ;
     88 000000                                            KEYWD$  "ACTIVE",       K$ACT
     89 000000                                            KEYWD$  "ADDRESS",      K$ADD
     90 000000                                            KEYWD$  "ADJACENCIES",  K$ADJ
     91 000000                                            KEYWD$  "ALL",          K$ALL
     92 000000                                            KEYWD$  "AREA",         K$ARE
     93 000000                                            KEYWD$  "ACCESS",       K$ACC
     94 000000                                            KEYWD$  "BILATERAL",    K$BIL
     95 000000                                            KEYWD$  "BLOCK",        K$BLK
     96 000000                                            KEYWD$  "BROADCAST",    K$BRO
     97 000000                                            KEYWD$  "BYTE",         K$BYT
     98 000000                                            KEYWD$  "BUFFER",       K$BUF
     99 000000                                            KEYWD$  "BUFFERS",      K$BUFS
    100 000000                                            KEYWD$  "CALL",         K$CALL
    101 000000                                            KEYWD$  "CHANNEL",      K$CHN
    102 000000                                            KEYWD$  "CHANNELS",     K$CHNS
    103 000000                                            KEYWD$  "CHARACTERISTICS",K$CHA
    104 000000                                            KEYWD$  "CIRCUIT",      K$CIR
    105 000000                                            KEYWD$  "CIRCUITS",     K$CIRS
    106 000000                                            KEYWD$  "CLEAR",        K$CLR
    107 000000                                            KEYWD$  "CLEARS",       K$CLRS
    108 000000                                            KEYWD$  "CLEARED",      K$CLE
    109 000000                                            KEYWD$  "CONSOLE",      K$CNSL
    110 000000                                            KEYWD$  "CONTROL",      K$CTL
```

```
                              ;    EVENT LIST
                              ;
000000                                   STATE$  EVELIS
000000                                    TRAN$  !EVERAN,$EXIT
000000                                    TRAN$  !EVENUM,$EXIT
                              ;
                              ;    EVENT RANGE
                              ;
000000                                   STATE$  EVERAN
000000                                    TRAN$  $DNUMB,,CKEVEB
000000                                   STATE$
000000                                    TRAN$  <'->
000000                                   STATE$
000000                                    TRAN$  $DNUMB,$EXIT,CKEVER
                              ;
                              ;    EVENT NUMBER
                              ;
000000                                   STATE$  EVENUM
000000                                    TRAN$  $DNUMB,$EXIT,CKEVEN

                                         .SBTTL  SUB-EXPRESSION TO PARSE LOGGING MONITOR

                              ;
                              ; PARSE LOGGING MONITOR
                              ;
000000                                   STATE$  LOGID
000000                                    TRAN$  &K$LOG

000000                                   STATE$  ,&P$LOG
000000                                    TRAN$  &K$MON,$EXIT,,OM.MON,$OPRND
000000                                    TRAN$  &K$FIL,$EXIT,,OM.FIL,$OPRND
000000                                    TRAN$  &K$CNSL,$EXIT,,OM.CON,$OPRND
                              ;
                              ; KNOWN LOGGING
                              ;
000000                                   STATE$  KNLOG
000000                                    TRAN$  &K$KNO

000000                                   STATE$  ,&P$PEN
000000                                    TRAN$  &K$LOG,$EXIT,,WC.LOG,$WILD

```

```
                        ;
000000                          STATE$  PLX
000000                           TRAN$  &K$DUP

000000                          STATE$  ,&P$DUP
000000                           TRAN$  &K$FUL, $EXIT,, SW.FUL, $SWICH
000000                           TRAN$  &K$HAL, $EXIT
                        ;
                        ; LOOK FOR BAUD RATE
                        ;
000000                          STATE$  BAUD
000000                           TRAN$  &K$SPD

000000                          STATE$  ,&P$SPD
000000                           TRAN$  !GT9600,$EXIT,,        -1,      $BAUD
000000                           TRAN$  $DNUMB, $EXIT,  .LISPD
                        ;
                        ; SUBEXPRESSION TO MATCH ">9600"
                        ;
000000                          STATE$  GT9600
000000                           TRAN$  RTBRK
000000                          STATE$
000000                           TRAN$  &K$9600,$EXIT
                        ;
                        ; LOOK FOR CSR
                        ;
000000                          STATE$  CSR
000000                           TRAN$  &K$CON, CSR1
000000                           TRAN$  $LAMDA

000000                          STATE$  CSR1
000000                           TRAN$  &K$CSR

000000                          STATE$  ,&P$CSR
000000                           TRAN$  $NUMBR, $EXIT,  .LICSR
                        ;
                        ; LOOK FOR SECONDARY CSR
                        ;
000000                          STATE$  UNCSR
000000                           TRAN$  &K$UNT, CSR2
000000                           TRAN$  $LAMDA                  ; HOPE IT GETS TO CONTROLLER FIRST

000000                          STATE$  CSR2
000000                           TRAN$  &K$CSR

000000                          STATE$  ,&P$CSR
000000                           TRAN$  $NUMBR, $EXIT,  .LISEC
                        ;
                        ; LOOK FOR UNIBUS RUN MASK
                        ;
000000                          STATE$  URM
000000                           TRAN$  &K$URM

000000                          STATE$  ,&P$URM
000000                           TRAN$  '[,     MASK

000000                          STATE$  NEXT
```

```
   2043 000000                               STATE$
   2044 000000                                TRAN$  '-,BYT1
   2045 000000                                TRAN$  $LAMDA, BYT2
   2046
   2047 000000                               STATE$  BYT1
   2048 000000                                TRAN$  &K$ARE, BYT2
   2049 000000                                TRAN$  $LAMDA
   2050
   2051 000000                               STATE$  BYT2,   &P$BYT
   2052 000000                                TRAN$  $DNUMB, $EXIT,   .SYBYT
   2053                               ;
   2054                               ; LOADING LOCATION
   2055                               ;
   2056 000000                               STATE$  LOC
   2057 000000                                TRAN$  &K$LOC, LOC1
   2058 000000                                TRAN$  $LAMDA, LOC2
   2059
   2060 000000                               STATE$  LOC1,   &P$LOC
   2061 000000                                TRAN$  $LAMDA
   2062
   2063 000000                               STATE$  LOC2
   2064 000000                                TRAN$  &K$TOP, $EXIT,,         SW.TOP, $SWICH
   2065 000000                                TRAN$  &K$FIR, $EXIT
   2066                               ;
   2067                               ; MAXIMUM CONTROL BUFFERS
   2068                               ;
   2069 000000                               STATE$  CCBNM
   2070 000000                                TRAN$  &K$MAX
   2071
   2072 000000                               STATE$  ,&P$MAX
   2073 000000                                TRAN$  &K$CTL
   2074
   2075 000000                               STATE$  ,&P$CON
   2076 000000                                TRAN$  &K$BUFS
   2077
   2078 000000                               STATE$  ,&P$CBN
   2079 000000                                TRAN$  $DNUMB, $EXIT,   .SYCBN
   2080                               ;
   2081                               ; MAXIMUM LARGE BUFFERS
   2082                               ;
   2083 000000                               STATE$  RDBNM
   2084 000000                                TRAN$  &K$MAX
   2085
   2086 000000                               STATE$  ,&P$MAX
   2087 000000                                TRAN$  &K$LAR
   2088
   2089 000000                               STATE$  ,&P$LAR
   2090 000000                                TRAN$  &K$BUFS
   2091
   2092 000000                               STATE$  ,&P$LBN
   2093 000000                                TRAN$  $DNUMB, $EXIT,   .SYRBN
   2094                               ;
   2095                               ; LARGE BUFFER SIZE
   2096                               ;
   2097 000000                               STATE$  RDBSZ
   2098 000000                                TRAN$  &K$LAR
   2099
```

```
ACOS2   010644R   002 CO.PRO= ****** GX     DXPRAB  012366R   002 E2.RET= ****** GX     K$CTL = 000026
AHOP2   010604R   002 CO.SYS= ****** GX     DXPRCH  012776R   002 E2.SEG= ****** GX     K$DAT = 000041
APR     005546R   002 CSR     006472R   002 DXPRC1  013010R   002 FILSPC  004352R   002 K$DEA = 000042
BAUD    006434R   002 CSR1    006500R   002 DXPRC2  013022R   002 FL.TRI= ****** GX     K$DEF = 000044
BLAUT   014666R   002 CSR2    006524R   002 EACOST  010626R   002 GNUM    014144R   002 K$DEV = 000047
BYT     011734R   002 CSTATE  005462R   002 EADD    010050R   002 GT9600  006464R   002 K$DFA = 000043
BYT1    011746R   002 CT      005612R   002 EAHOPS  010566R   002 HBT     007042R   002 K$DIAG= 000050
BYT2    011754R   002 CTM     013462R   002 EAMAX   010534R   002 HEXDIG  014502R   002 K$DIS = 000051
CCBNM   012024R   002 CUG     000744R   002 EBEA    010740R   002 HEXSTR  014450R   002 K$DMP = 000054
CHN     005574R   002 DEFCIR  005020R   002 EBEA1   010754R   002 HEXST2  014466R   002 K$DST = 000045
CHNL    014176R   002 DEFEXE  007444R   002 EBEA2   010770R   002 HOSNAM  003260R   002 K$DSTS= 000046
CHNL1   014212R   002 DEFINE  001030R   002 EBEA3   010772R   002 HRDSTR  004034R   002 K$DTE = 000052
CIMXR   005334R   002 DEFLIN  006024R   002 EBRT    010666R   002 HRDST2  004040R   002 K$DTES= 000053
CIMXR2  005352R   002 DEFLOG  002350R   002 EBRT2   010704R   002 HTM     005640R   002 K$DUP = 000055
CIMXR3  005360R   002 DEFMOD  012252R   002 EBRT3   010720R   002 IN$CHA= ****** GX     K$ENA = 000056
CIRC    005042R   002 DEFNOD  002736R   002 ECOST   010342R   002 IN$SUM= ****** GX     K$EVE = 000057
CIRID   007172R   002 DEFOBJ  011210R   002 EHOP    010314R   002 KNCIR   007424R   002 K$EXE = 000060
CIRI1   007224R   002 DEFPRO  004556R   002 EHOST   010070R   002 KND     000762R   002 K$EXI = 000061
CIROPT  005066R   002 DEFSYS  011476R   002 EID     010156R   002 KNDS    000776R   002 K$FAC = 000062
CIRPR   005374R   002 DEFXAC  012462R   002 EID1    010164R   002 KNG     000730R   002 K$FIL = 000063
CIRPR2  005404R   002 DEFXPR  012350R   002 EINAC   011050R   002 KNLIN   007404R   002 K$FIR = 000064
CIRO    005054R   002 DEFXSV  012402R   002 EINA2   011060R   002 KNLOG   002716R   002 K$FUL = 000065
CISER   005420R   002 DEFX29  012532R   002 EINCT   011100R   002 KNNOD   004536R   002 K$GRP = 000066
CKEVEB= ****** GX     DFB     013536R   002 EINC2   011110R   002 KNOBJ   011456R   002 K$GRPS= 000067
CKEVEC= ****** GX     DFW     013564R   002 ELINK   010130R   002 KNPRO   004762R   002 K$HAL = 000071
CKEVEN= ****** GX     DFXAC1  012474R   002 ENAME   010032R   002 K$ACC = 000005        K$HEL = 000073
CKEVER= ****** GX     DFXAC2  012506R   002 ENCOU   010416R   002 K$ACT = 000000        K$HLD = 000075
CKFSP = ****** GX     DFXSV1  012424R   002 ENN     010370R   002 K$ADD = 000001        K$HLO = 000074
CLM     014370R   002 DFXSV2  012436R   002 EOUTT   011130R   002 K$ADJ = 000002        K$HOP = 000076
CLT     013510R   002 DFX29A  012554R   002 EOUT2   011140R   002 K$ALL = 000003        K$HOS = 000077
CLV     014420R   002 DFX29B  012566R   002 ERETF   011160R   002 K$ARE = 000004        K$HRD = 000072
CM.DEF= ****** GX     DNUM    014234R   002 ERET2   011170R   002 K$BIL = 000006        K$ID  = 000101
CM.EXI= ****** GX     DNUM0   014254R   002 EROU    010454R   002 K$BLK = 000007        K$IDE = 000100
CM.HLP= ****** GX     DNUM1   014262R   002 ERRFSP= ****** GX     K$BRO = 000010        K$INA = 000102
CM.KIL= ****** GX     DPR     006766R   002 ERRILN= ****** GX     K$BUF = 000012        K$INC = 000103
CM.LIS= ****** GX     DPRI    014536R   002 ERRIPV= ****** GX     K$BUFS= 000013        K$INS = 000104
CM.PUR= ****** GX     DQUOTE  010254R   002 ESEG    011012R   002 K$BYT = 000011        K$KIL = 000105
CNTMAX  004734R   002 DST     001012R   002 ESEG2   011022R   002 K$CALL= 000014        K$KNO = 000106
COPY    011332R   002 DSTA    014120R   002 ESEG3   011030R   002 K$CHA = 000017        K$LAR = 000107
COST    005512R   002 DSTLIS  012612R   002 ESUB    010502R   002 K$CHN = 000015        K$LIN = 000110
CO.CIR= ****** GX     DSTLS0  012656R   002 ESUB2   010526R   002 K$CHNS= 000016        K$LINS= 000111
CO.EXE= ****** GX     DSTLS1  012634R   002 EVEID   002542R   002 K$CIR = 000020        K$LIS = 000113
CO.KLG= ****** GX     DSTLS2  012644R   002 EVEID1  002572R   002 K$CIRS= 000021        K$LNKS= 000112
CO.KNC= ****** GX     DSUB    014554R   002 EVEID2  002612R   002 K$CLE = 000024        K$LOAD= 000114
CO.KNL= ****** GX     DSUB2   014600R   002 EVELIS  002620R   002 K$CLR = 000022        K$LOC = 000115
CO.KNN= ****** GX     DTE     014276R   002 EVENUM  002650R   002 K$CLRS= 000023        K$LOG = 000116
CO.KNO= ****** GX     DTENUM  014300R   002 EVERAN  002634R   002 K$CNSL= 000025        K$LOGI= 000117
CO.KNP= ****** GX     DTE0    014316R   002 EVFY    010262R   002 K$CON = 000027        K$LOO = 000120
CO.LIN= ****** GX     DTE1    014322R   002 EXEOPT  007476R   002 K$COP = 000030        K$MAX = 000122
CO.LOG= ****** GX     DXPCG0  013400R   002 EXE0    007464R   002 K$COS = 000031        K$MIN = 000123
CO.MAC= ****** GX     DXPCG1  013412R   002 EXIT    015056R   002 K$COU = 000032        K$MOD = 000124
CO.MSP= ****** GX     DXPCG2  013424R   002 EXPUR   000120R   002 K$COUN= 000033        K$MON = 000125
CO.MXP= ****** GX     DXPCUG  013350R   002 E2.BEA= ****** GX     K$CPA = 000034        K$MSK = 000121
CO.MXS= ****** GX     DXPDDT  013212R   002 E2.BRT= ****** GX     K$CPB = 000035        K$MUL = 000126
CO.MX9= ****** GX     DXPDD0  013242R   002 E2.INA= ****** GX     K$CPC = 000036        K$NAM = 000127
CO.NOD= ****** GX     DXPDD1  013254R   002 E2.INC= ****** GX     K$CPD = 000037        K$NET = 000130
CO.OBJ= ****** GX     DXPDD2  013266R   002 E2.OUT= ****** GX     K$CSR = 000040        K$NOD = 000131
```

```
                        ;****
                        ; LOCAL MACROS
                        ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT

                                .MACRO  ERROR$  TEXT
                        ;
                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                        ; MESSAGE STRING.
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                .ENDC
                                CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                .ENDM   ERROR$

                                .MACRO  EPRINT  TEXT
                        ;
                        ; PRINT TEXT ON ERROR LUN
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                .ENDC
                                CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                .ENDM   EPRINT


                        ;
                        ; MACRO LIBRARY CALLS
                        ;

                                .MCALL  EXIT$S,DELET$,CLOSE$,EXST$S,QIOW$,DIR$
                                .MCALL  NBOF$L,FDOF$L

 000000                         FDOF$L          ; FDB OFFSETS
 000000                         NBOF$L          ; NAME BLOCK OFFSETS

                        ;
                        ; LOCAL DATA
                        ;

 000000                 WATDPB: QIOW$   IO.WAT,0,1,,IOSB,,<$OFDB+F.FNB+N.FID,$FIATR>
 000030                 IOSB:   .BLKW   2

```

```
000252  005702                                  TST     R2              ;FIRST/BEST FIT?
000254  100420                                  BMI     6$              ;IF MI FIRST FIT
000256  005705                                  TST     R5              ;FIND PREVIOUS BLOCK?
000260  001404                                  BEQ     3$              ;IF EQ NO
000262  026563  000002  000002                  CMP     2(R5),2(R3)     ;COMPARE BLOCK SIZES
000270  101402                                  BLOS    4$              ;IF LOS PREVIOUS WINS
000272  010004                          3$:     MOV     R0,R4           ;SET ADDR OF PREV
000274  010305                                  MOV     R3,R5           ;SET ADDR OF BLOCK
000276  010300                          4$:     MOV     R3,R0           ;SET NEW PREV ADDR
000300  000756                                  BR      2$              ;GO AGAIN
000302  005702                          5$:     TST     R2              ;FIRST/BEST FIT?
000304  100416                                  BMI     9$              ;IF MI FIRST FIT
000306  005705                                  TST     R5              ;FIND A BLOCK?
000310  001414                                  BEQ     9$              ;IF EQ NO
000312  010400                                  MOV     R4,R0           ;RETRIEVE ADDR OF PREV
000314  010503                                  MOV     R5,R3           ;RETRIEVE ADDR OF BLOCK
000316  160163  000002                  6$:     SUB     R1,2(R3)        ;REDUCE SIZE OF BLOCK
000322  001403                                  BEQ     7$              ;IF EQ NONE LEFT
000324  066303  000002                          ADD     2(R3),R3        ;CALCULATE ADDRESS OF BLOCK
000330  000401                                  BR      8$              ;
000332  011310                          7$:     MOV     (R3),(R0)       ;LINK PREV TO NEXT
000334  010300                          8$:     MOV     R3,R0           ;SET ADDRESS OF BLOCK
000336  000241                                  CLC                     ;CLEAR CARRY
000340  000401                                  BR       10$            ;
000342  000261                          9$:     SEC                     ;SET CARRY
000344  000207                          10$:    RETURN                  ;

        000001                                  .END                    ;
```

```
                                                                  G 14
CFESTB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE             REFERENCES

CM.EXI  = ******  GX       6-171
ERRALL  = ******  GX       8-240
ERRSYN  = ******  GX       6-186
RBRKT   = 000076          #4-99         6-196       6-197
SEMI    = 000073          #4-98
TMPKTB    000000 RG       #5-106        6-178
TMPSTB    000000 RG       #5-106        6-180
$ALPHA  = 000022          #5-106
$ANY    = 000020          #5-106
$BLANK  = 000006          #5-106
$BUFI   = ******  GX       6-179        6-198
$CFERR  = ******  GX       6-186        8-240
$CFER1  = ******  GX       6-198
$CMAND  = ******  GX      *6-169        6-171
$DIGIT  = 000024          #5-106
$DNUMB  = 000014          #5-106
$EOS    = 000012          #5-106
$ERROR    000000 RG       #4-100       *6-170       6-183
$EXIT   = 000000          #5-106
$FAIL   = 177777          #5-106
$GPRM   = ******           5-106
$LAMDA  = 000000          #5-106
$NUMBR  = 000002          #5-106
$QINS   = ******  GX       7-220
$RAD50  = 000016          #5-106
$RDATA  = ******  GX       6-173
$RONLY  = ******           5-106        5-106       5-106
$RQBF   = ******  GX       8-238
$STRNG  = 000004          #5-106
$SUBXP  = 000010          #5-106
$TALOC    000146 RG       #7-216
$TMLST  = ******  GX       7-219
$TMSTB    000002 RG       #6-168
$T1ALC    000170 RG        7-218       #8-235
$XXAVL  = ******  GX       7-217        8-236
$$$FLG  = 177777          #5-106
$$$KEY  = 177777          #5-106
.EXIT   = ******  GX       8-241
.TPARS  = ******  GX       6-181
```

```
                                                                  H 14
CFESTB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 2
```

```
    332 000362  062700  000011          5$:     ADD     #22-11,R0                       ; ELSE, PERIOD, SLASH OR NUMBER (+22)
    333 000366  062700  177711          10$:    ADD     #11-100,R0                      ; DOLLAR (+11)
    334 000372  062700  000040          20$:    ADD     #100-40,R0                      ; ALPHA (+100)
    335 000376  062700  000040          30$:    ADD     #40,R0                          ; BLANK (40)
    336 000402  000207                          RETURN
```

```
                                                                          G 16
CFESUB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

E$NUSE    000004            #4-63
E$STRT    000006            #4-63
FF.ADD  = 000020            #5-95       5-95
FF.CIR  = 000040            #5-95       5-95
FF.CON  = 000001            #5-95
FF.FIL  = 000002            #5-95
FF.HST  = 040000            #5-95
FF.LIN  = 000010            #5-95       5-95
FF.MOD  = 000100            #5-95       5-95
FF.MON  = 000004            #5-95
FF.MSK  = 000077            #5-95
FF.PRT  = 000200            #5-95       5-95
FF.QUL  = 000370            #5-95
FF.REM  = 100000            #5-95
FL.MFL  = ******   GX        14-502     14-505
FL.MTP  = ******   GX        14-498     14-501
FL.MUX  = ******   GX        14-510
FL.TRI  = ******   GX        14-518
F.ADD     000016            #5-95
F.CEV     000030            #5-95
F.CIR     000024            #5-95
F.CLS     000002            #5-95
F.EVT     000004            #5-95
F.FLG     000014            #5-95
F.LEN     000016            #5-95       5-95
F.LENX  = 000020   G        #5-95
F.LIN     000020            #5-95
F.LNK     000000            #5-95
F.MOD     000022            #5-95
F.REM     000026            #5-95
F.SEV     000004            #5-95
G$LEN     000016             5-98
G$LENX  = 000020   G        #5-98
IO.WVB  = ******   GX        7-161
LF.MFL  = ******   GX        14-503
LF.MTP  = ******   GX        14-499     14-516
L$CNT   = ******   GX        14-508
L$LEN     000122             5-96
L$LENX  = 000124   G        #5-96
L$NAM   = ******   GX        14-496
L$TFI   = ******   GX        15-553
L$UNT   = ******   GX        14-514
L.LNG     000150             5-111
L.LNGX  = 000150   G        #5-111
N$$SLI  = 000001            #5-110
O.LEN     000012             5-93
O.LENX  = 000014   G        #5-93
PRINT     000014 R          #7-164      8-204
Q.IOFN  = ******   GX       *7-161
Q.IOPL  = ******   GX       *7-165     *7-169       7-170      *7-172      *7-174      *8-[illegible]     *8-210       8-211      *8-213
                            *8-215
RF.L00  = 100000            #5-94
```

```
                                                                          H 16
CFESUB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 3
```

```
        002000                          OC.HTM  == 2000          ; HELLO TIMER
        004000                          OC.MXR  == 4000          ; MAXIMUM ROUTERS       (ETHERNET)
        010000                          OC.APR  == 10000         ; ACTIVE POLLING RATIO PRESENT
        020000                          OC.USG  == 20000         ; USAGE                 (PSI)
        040000                          OC.RPR  == 40000         ; ROUTER PRIORITY       (ETHERNET)
        100000                          OC.NUM  == 100000        ; NUMBER                (PSI, SVC ONLY)
                                ;
                                ; NO MORE BITS AVAILABLE:  SEE SW.xxx FOR ADDITIONAL OPTIONS.
                                ;

        002000                            OC.EXE        == OC.HTM                           ; TRANSPORT ONLY
        000060                            OC.PVC        == OC.CHN!OC.DTE                    ; PVC ONLY
        101400                            OC.SVC        == OC.RTM!OC.MRC!OC.NUM             ; SVC ONLY
        101570                            OC.X25        == OC.PVC!OC.SVC!OC.DTE!OC.MBL!OC.MWI ; X25 ONLY
        046002                            OC.SLT        == OC.STA!OC.HTM!OC.MXR!OC.RPR      ; SLT PARAMETERS
        000000                            OC.CNT        == 0                                ; CONTROLLER PARAMETERS
        010204                            OC.STN        == OC.APR!OC.CST!OC.TRI             ; STATION PARAMETERS
        010204                            OC.UNT        == OC.STN                           ; UNITS/STATIONS
        000240                            OC.NWC        == OC.TRI!OC.CHN                    ; NO WILDCARDS ON THESE

                                        .IIF NE <OC.CNT & OL.CNT> .ERROR LINE/CIRCUIT OPTIONS NOT UNIQUE
                                        .IIF NE <OC.STN & OL.STN> .ERROR LINE/CIRCUIT OPTIONS NOT UNIQUE
                                        .IIF NE <OC.UNT & OL.UNT> .ERROR LINE/CIRCUIT OPTIONS NOT UNIQUE


                                ;
                                ; PROCESS OPTIONS
                                ;
        000001                          OP.STA == 1              ; =1 IF PROCESS STATE IS PRESENT
        000002                          OP.NLT == 2              ; =1 IF NUMBER OF LINE TABLES PRESENT
        000004                          OP.NCT == 4              ; =1 IF NUMBER OF CONTROLLER TABLES PRESENT
                                ;
                                ; EXECUTOR OPTIONS
                                ;
        000001                          OE.NAM == 1              ; =1 IF NODE NAME IS PRESENT
        000002                          OE.ADD == 2              ; =1 IF NODE ADDRESS PRESENT
        000004                          OE.ID  == 4              ; =1 IF NODE ID IS PRESENT
        000010                          OE.LNK == 10             ; =1 IF LOGICAL LINK COUNT PRESENT
        000020                          OE.MXC == 20             ; =1 IF THE ROUTING COST MAXIMUM IS SPECIFIED
        000040                          OE.MXH == 40             ; =1 IF THE ROUTING HOP MAXIMUM IS SPECIFIED
        000100                          OE.NN  == 100            ; =1 IF THE ROUTING MAXIMUM ADDRESS IS SPECIFIED
        000200                          OE.NCO == 200            ; =1 IF MAXIMUM NODE COUNTERS IS SPECIFIED
        000400                          OE.VFY == 400            ; =1 IF VERIFICATION STATE PRESENT
        001000                          OE.HOS == 1000           ; =1 IF HOST ADDRESS PRESENT
        002000                          OE.RTM == 2000           ; =1 IF ROUTING TIMER PRESENT

        004000                          OE.SUB == 4000           ; =1 IF TRANSPORT SUBADDRESS(ES) PRESENT

        010000                          OE.AMAX == 10000         ; =1 IF MAXIMUM AREA PRESENT
        020000                          OE.AHOP == 20000         ; =1 IF AREA MAX HOPS PRESENT
        040000                          OE.ACOS == 40000         ; =1 IF AREA MAX COST PRESENT
                                ;
                                ; EXEC FLAGS: $OPRN2
                                ;
        000001                          E2.BRT  == 1             ; BROADCASI ROUTING TIMER PRESENT
        000002                          E2.BEA  == 2             ; BROADCAST ENDNODE ADJ PRESENT
        000004                          E2.SEG  == 4             ; SEGMENT BUFFER SIZE PRESENT
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE              REFERENCES

A.LULU  = 000002              6-95
BUFSIZ  = 000764             #5-74         6-95          6-112         7-130
CH$HI   = 000400   G         #12-1163
CL.AST  = 004000             #6-95
CL.ATT  = 000200              6-95        #6-95
CL.CAS  = 020000             #6-95
CL.CMT  = 000001   G          6-95        #6-95          6-96
CL.EIF  = 000040   G         #6-95         6-96
CL.EXN  = 040000             #6-95
CL.EXT  = 000010              6-95        #6-95
CL.IND  = 000002              6-95        #6-95
CL.LCC  = 000020   G          6-95        #6-95          6-96
CL.NPR  = 000400             #6-95
CL.OPN  = 000100   G         #6-95         6-96
CL.RNE  = 000004   G         #6-95         6-96
CL.XIT  = 100000   G         #6-95         6-96
CM.DEF  = 000001   G         #11-695
CM.EXI  = 000004   G         #11-698
CM.HLP  = 000003   G         #11-697
CM.KIL  = 000005   G         #11-699
CM.LIS  = 000002   G         #11-696
CM.PUR  = 000006   G         #11-700
CO.CIR  = 000010   G         #11-718
CO.EXE  = 000003   G         #11-710
CO.KLG  = 000007   G         #11-717
CO.KNC  = 000011   G         #11-719
CO.KNL  = 000006   G         #11-715
CO.KNN  = 000001   G         #11-707
CO.KNO  = 000005   G         #11-713
CO.KNP  = 000002   G         #11-709
CO.LIN  = 000006   G         #11-714
CO.LOG  = 000007   G         #11-716
CO.MAC  = 000014   G         #11-722
CO.MSP  = 000016   G         #11-724
CO.MXP  = 000012   G         #11-720
CO.MXS  = 000013   G         #11-721
CO.MX9  = 000015   G         #11-723
CO.NOD  = 000001   G         #11-706
CO.OBJ  = 000005   G         #11-712
CO.PRO  = 000002   G         #11-708
CO.SYS  = 000004   G         #11-711
CS.BUF  = 000001   G         #12-1093
CS.CHN  = 000027   G         #12-1118
CS.CNT  = 000006   G         #12-1098
CS.CUG  = 000031   G         #12-1120
CS.DDM  = 000005   G         #12-1097
CS.DEC  = 000046   G         #12-1144
CS.DLC  = 000004   G         #12-1096
CS.DSA  = 000033   G         #12-1122
CS.DSC  = 000034   G         #12-1123
CS.DST  = 000032   G         #12-1121
CS.DTE  = 000026   G         #12-1117
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 14
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE            REFERENCES

$MRC     = 001560 RG      #7-241
$MRE     = 001550 RG      #7-254
$MWI     = 001556 RG      #7-242
$MXC     = 001560 RG      #7-253
$MXNCT   = 000100  G      #5-84
$MXNLT   = 000100  G      #5-81
$MXR     = 001570 RG      #7-256
$MXSTA   = 000100  G      #5-80
$NCOU    = 001574 RG      #7-222
$NCT     = 001620 RG      #7-224
$NLN       001706 RG      #7-216
$NLT     = 001564 RG      #7-223
$NN      = 001556 RG      #7-221      7-242      7-246
$NNAM      001724 RG      #7-259
$NODAD     001546 RG      #7-190
$NODNA     001542 RG      #7-189      7-257
$NTADD     001504 RG      #7-161
$NTNAM     001500 RG      #7-160
$NUM     = 001550 RG      #7-258
$OBJ       001712 RG      #7-218      7-232
$OFDB      000002 RG      #6-104
$OPRND     001452 RG      #7-144
$OPRN2     001454 RG      #7-145
$PARNA     001570 RG      #7-198      7-229      7-230
$PBLK      001516 RG      #7-169
$POLNA     001564 RG      #7-197      7-223      7-226      7-240      7-247
$PRI       001552 RG      #7-192      7-220
$PRNAM     001532 RG      #7-181      7-182
$PSHPT     000174 RG      #6-95
$PSIPT     001424 RG      #7-133
$RDBNM     001522 RG      #7-174
$RDBSZ     001524 RG      #7-175
$RET     = 001620 RG      #7-255
$RNDCP   = 001562 RG      #7-312      7-313
$RNDIA   = 001755 RG      #7-319      7-320
$RNDMP   = 002014 RG      #7-320      7-321
$RNDPA   = 001556 RG      #7-311      7-312
$RNHAR   = 001574 RG      #7-314      7-315
$RNHOS   = 001552 RG      #7-309      7-310
$RNLOD   = 001716 RG      #7-318      7-319
$RNSCI   = 001564 RG      #7-313      7-314
$RNSDV   = 001554 RG      #7-310      7-311
$RNSLO   = 001620 RG      #7-316      7-317
$RNSPA   = 001604 RG      #7-315      7-316
$RNTLO   = 001657 RG      #7-317      7-318
$RNXXX   = 002053 RG      #7-321      7-323      7-323
$RST       001574 RG      #7-201
$RTC     = 001616 RG      #7-245
$RTM       001722 RG      #7-243
$SACNM   = 001734 RG      #7-281      7-282
$SCIRC   = 001724 RG      #7-280      7-281
$SCSR      001560 RG      #7-195      7-231      7-241      7-253
```

Table of contents

```
     5-    95    $CFLIS - PROCESS LIST COMMAND
```

```
                                        ;+
                                        ; *** - $WDATA - WRITE DATA TO OUTPUT FILE
                                        ;
                                        ; INPUTS:
                                        ;       INITIALIZED FDB AT $OFDB
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT=SUCCESS/FAILURE
                                        ;       R0 = ADDRESS OF FDB
                                        ;
                                        ;-

000662                                  $WDATA::
000662  012700  000000G                         MOV     #$OFDB,R0       ; POINT AT FDB
000666                                          PUT$                    ; WRITE THE RECORD
000672  000207                                  RETURN


        000001                                  .END
```

```
000574  001412                            BEQ     60$                     ; BR IF NO
000576  012700  177777                    MOV     #177777,R0              ; INDICATE ALL EVENTS
000602  010067  000000G                   MOV     R0,$EVENT+0             ; ...
000606  010067  000002G                   MOV     R0,$EVENT+2             ; ...
000612  010067  000004G                   MOV     R0,$EVENT+4             ; ...
000616  010067  000006G                   MOV     R0,$EVENT+6             ; ...
000622  012700  000000G           60$:    MOV     #$EVENT,R0              ; POINT TO SPECIFIED EVENT MASKS

        000004                            .REPT   4
                                          BIC     (R0)+,(R1)+             ; CLEAR SPECIFIED EVENT MASKS
                                          .ENDR

        000004                            .REPT   4
                                          TST     -(R1)                   ; ANY EVENT MASKS SET?
                                          BNE     70$                     ; BR IF YES
                                          .ENDR

000656  004767  000362                    CALL    QREM                    ; UNLINK TEMPLATE FROM LIST
000662  012701  000020                    MOV     #EVTLEN,R1              ; GET LENGTH OF EVENT TEMPLATE
000666  016700  000000G                   MOV     $XXAVL,R0               ; POINT AT END OF TASK LISTHEAD
000672  004767  000000G                   CALL    $RLBF                   ; RELEASE CORE BLOCK
000676  162767  000000G 000000G           SUB     #F.LENX,$BYTMN          ; ADJUST MINIMUM POOL BYTE EXTENSION
000704  162767  000000G 000000G           SUB     #F.LENX,$BYTXT          ; AND CURRENT BYTE EXTENSION VALUE
000712  032767  000000G 000000G   70$:    BIT     #WC.LOG,$WILD           ; KNOWN LOGGING SPECIFIED?
000720  001275                            BNE     40$                     ; BR IF YES
000722  032767  000000G 000000G   75$:    BIT     #OM.KEV,$OPRND          ; KNOWN EVENTS SPECIFIED?
000730  001225                            BNE     10$                     ; BR IF YES
000732  000207                    80$:    RETURN
```

```
    111 000000                                            KEYWD$  "CONTROLLER",    K$CON
    112 000000                                            KEYWD$  "COPIES",        K$COP
    113 000000                                            KEYWD$  "COST",          K$COS
    114 000000                                            KEYWD$  "COUNTERS",      K$COU
    115 000000                                            KEYWD$  "COUNT",         K$COUN
    116 000000                                            KEYWD$  "CPA",           K$CPA
    117 000000                                            KEYWD$  "CPB",           K$CPB
    118 000000                                            KEYWD$  "CPC",           K$CPC
    119 000000                                            KEYWD$  "CPD",           K$CPD
    120 000000                                            KEYWD$  "CSR",           K$CSR
    121 000000                                            KEYWD$  "DATA",          K$DAT
    122 000000                                            KEYWD$  "DEAD",          K$DEA
    123 000000                                            KEYWD$  "DEFAULT",       K$DFA
    124 000000                                            KEYWD$  "DEFINE",        K$DEF
    125 000000                                            KEYWD$  "DESTINATION",   K$DST
    126 000000                                            KEYWD$  "DESTINATIONS",  K$DSTS
    127 000000                                            KEYWD$  "DEVICE",        K$DEV
    128 000000                                            KEYWD$  "DIAGNOSTICS",   K$DIAG
    129 000000                                            KEYWD$  "DISABLED",      K$DIS
    130 000000                                            KEYWD$  "DTE",           K$DTE
    131 000000                                            KEYWD$  "DTES",          K$DTES
    132 000000                                            KEYWD$  "DUMP",          K$DMP
    133 000000                                            KEYWD$  "DUPLEX",        K$DUP
    134 000000                                            KEYWD$  "ENABLED",       K$ENA
    135 000000                                            KEYWD$  "EVENTS",        K$EVE
    136 000000                                            KEYWD$  "EXECUTOR",      K$EXE
    137 000000                                            KEYWD$  "EXIT",          K$EXI
    138 000000                                            KEYWD$  "FACTOR",        K$FAC
    139 000000                                            KEYWD$  "FILE",          K$FIL
    140 000000                                            KEYWD$  "FIRSTFIT",      K$FIR
    141 000000                                            KEYWD$  "FULL",          K$FUL
    142 000000                                            KEYWD$  "GROUP",         K$GRP
    143 000000                                            KEYWD$  "GROUPS",        K$GRPS
    144 000000                                            KEYWD$  "9600",          K$9600
    145 000000                                            KEYWD$  "HALF",          K$HAL
    146 000000                                            KEYWD$  "HARDWARE",      K$HRD
    147 000000                                            KEYWD$  "HELP",          K$HEL
    148 000000                                            KEYWD$  "HELLO",         K$HLO
    149 000000                                            KEYWD$  "HOLDBACK",      K$HLD
    150 000000                                            KEYWD$  "HOPS",          K$HOP
    151 000000                                            KEYWD$  "HOST",          K$HOS
    152 000000                                            KEYWD$  "IDENTIFICATION",K$IDE
    153 000000                                            KEYWD$  "ID",            K$ID
    154 000000                                            KEYWD$  "INACTIVITY",    K$INA
    155 000000                                            KEYWD$  "INCOMING",      K$INC
    156 000000                                            KEYWD$  "INSPECT",       K$INS
    157 000000                                            KEYWD$  "KILL",          K$KIL
    158 000000                                            KEYWD$  "KNOWN",         K$KNO
    159 000000                                            KEYWD$  "LARGE",         K$LAR
    160 000000                                            KEYWD$  "LINE",          K$LIN
    161 000000                                            KEYWD$  "LINES",         K$LINS
    162 000000                                            KEYWD$  "LINKS",         K$LNKS
    163 000000                                            KEYWD$  "LIST",          K$LIS
    164 000000                                            KEYWD$  "LOADER",        K$LOAD
    165 000000                                            KEYWD$  "LOCATION",      K$LOC
    166 000000                                            KEYWD$  "LOGGING",       K$LOG
    167 000000                                            KEYWD$  "LOGIN",         K$LOGI
```

```
                                .SBTTL  DEFINE NODE COMMAND
                        ;
                        ; DEFINE NODE COMMAND
                        ;
000000                          STATE$  DEFNOD
000000                           TRAN$  !NODID,,,                   CO.NOD, $COMP

000000                          STATE$  ,&P$DENO
000000                           TRAN$  &K$NAM, NODNAM
000000                           TRAN$  $LAMDA, RNDLL

000000                          STATE$  NODNAM,&P$NAM
000000                           TRAN$  $STRNG,,            .NONAM, RN.NAM, $OPRND  ;Remote node name
                        ;
                        ; Parameters for remote nodes
                        ;
000000                          STATE$  RNDLL
000000                           TRAN$  $LAMDA,,            .RNIN2
000000                          STATE$  RNOPT
000000                           TRAN$  !RNHOS, RNOPT,,             RN.HOS, $OPRND  ;Host node ID
000000                           TRAN$  !RNSDV, RNOPT,,             RN.SDV, $OPRND  ;Service device
000000                           TRAN$  !RNSNV, RNOPT,,             RN.SNV, $OPRND  ;Service node version
000000                           TRAN$  !RNDPA, RNOPT,,             RN.DPA, $OPRND  ;Dump address
000000                           TRAN$  !RNDCT, RNOPT,,             RN.DCT, $OPRND  ;Dump count
000000                           TRAN$  !RNSCI, RNOPT,,             RN.SCI, $OPRND  ;Service circuit
000000                           TRAN$  !RNHAR, RNOPT,,             RN.HAR, $OPRND  ;Hardware address
000000                           TRAN$  !RNSPA, RNOPT,,             RN.SPA, $OPRND  ;Service password
000000                           TRAN$  !RNSLO, RNOPT,,             RN.SLO, $OPRND  ;Secondary loader file
000000                           TRAN$  !RNTLO, RNOPT,,             RN.TLO, $OPRND  ;Tertiary loader file
000000                           TRAN$  !RNLOD, RNOPT,,             RN.LOD, $OPRND  ;Load file
000000                           TRAN$  !RNDIA, RNOPT,,             RN.DIA, $OPRND  ;Diagnostics file
000000                           TRAN$  !RNDMP, RNOPT,,             RN.DMP, $OPRND  ;Dump file
000000                           TRAN$  $EOS,    $EXIT,     .RNCHK
                        ;
                        ; Host node ID
                        ;
000000                          STATE$  RNHOS
000000                           TRAN$  &K$HOS
000000                          STATE$  ,&P$HOS
000000                           TRAN$  $LAMDA,,            .RNHO1
000000                          STATE$
000000                           TRAN$  !NODADD,$EXIT,      .RNHOS
000000                           TRAN$  !HOSNAM,$EXIT,      .RNHOS
000000                           TRAN$  $LAMDA, $FAIL,      .RNHO2

000000                          STATE$  HOSNAM
000000                           TRAN$  $STRNG, $EXIT       .RNHNA
000000                           TRAN$  $LAMDA, $FAIL       .RNHO2
                        ;
                        ; Service device
                        ;
000000                          STATE$  RNSDV
000000                           TRAN$  &K$DEV, RNSDV2
000000                           TRAN$  &K$SER
000000                          STATE$  ,&P$SERO
000000                           TRAN$  &K$DEV
000000                          STATE$  RNSDV2,&P$SDV
```

```
000000                          TRAN$   <',>,     MASK
000000                          TRAN$   '],       $EXIT

000000                         STATE$   MASK
000000                          TRAN$   &K$CPA,NEXT,,1,$URM
000000                          TRAN$   &K$CPB,NEXT,,2,$URM
000000                          TRAN$   &K$CPC,NEXT,,4,$URM
000000                          TRAN$   &K$CPD,NEXT,,10,$URM
000000                          TRAN$   &K$UBE,NEXT,,20,$URM
000000                          TRAN$   &K$UBF,NEXT,,40,$URM
000000                          TRAN$   &K$UBH,NEXT,,100,$URM
000000                          TRAN$   &K$UBJ,NEXT,,200,$URM
000000                          TRAN$   &K$UBK,NEXT,,400,$URM
000000                          TRAN$   &K$UBL,NEXT,,1000,$URM
000000                          TRAN$   &K$UBM,NEXT,,2000,$URM
000000                          TRAN$   &K$UBN,NEXT,,4000,$URM
000000                          TRAN$   &K$UBP,NEXT,,10000,$URM
000000                          TRAN$   &K$UBR,NEXT,,20000,$URM
000000                          TRAN$   &K$UBS,NEXT,,40000,$URM
000000                          TRAN$   &K$UBT,NEXT,,100000,$URM
                        ;
                        ; MULTI-POINT DEAD RATIO
                        ;
000000                         STATE$   DPR
000000                          TRAN$   &K$MUL

000000                         STATE$   ,&P$MUL
000000                          TRAN$   &K$DEA

000000                         STATE$   ,&P$DEA
000000                          TRAN$   $DNUMB,$EXIT,.LIDEA
                        ;
                        ; LOOK FOR COUNTER TIMER
                        ;
000000                         STATE$   LCT
000000                          TRAN$   &K$COU
000000                         STATE$   ,&P$CTO
000000                          TRAN$   &K$TIM
000000                         STATE$   ,&P$CT
000000                          TRAN$   $DNUMB,$EXIT,.LICTM
                        ;
                        ; LOOK FOR HOLDBACK TIMER
                        ;
000000                         STATE$   HBT
000000                          TRAN$   &K$HLD
000000                         STATE$   ,&P$HBO
000000                          TRAN$   &K$TIM
000000                         STATE$   ,&P$HBT
000000                          TRAN$   $DNUMB,$EXIT,.LIHBT
                        ;
                        ; MAXIMUM RETRANSMITS
                        ;
000000                         STATE$   MRT
000000                          TRAN$   &K$MAX
000000                         STATE$   ,&P$MAX
000000                          TRAN$   &K$RTRS
000000                         STATE$   ,&P$RTC
```

DEFINE SYSTEM COMMAND

```
 000000                                   STATE$  ,&P$LAR
 000000                                    TRAN$  &K$BUF

 000000                                   STATE$  ,&P$BUF
 000000                                    TRAN$  &K$SIZ

 000000                                   STATE$  ,&P$LBS
 000000                                    TRAN$  $DNUMB, $EXIT,  .SYRBS
                                ;
                                ; MAXIMUM SMALL BUFFERS
                                ;
 000000                                   STATE$  SDBNM
 000000                                    TRAN$  &K$MAX

 000000                                   STATE$  ,&P$MAX
 000000                                    TRAN$  &K$SMA

 000000                                   STATE$  ,&P$SMA
 000000                                    TRAN$  &K$BUFS

 000000                                   STATE$  ,&P$SBN
 000000                                    TRAN$  $DNUMB, $EXIT,  .SYSBN
                                ;
                                ; RECEIVE BUFFER MINIMUM
                                ;
 000000                                   STATE$  THRESH
 000000                                    TRAN$  &K$MIN

 000000                                   STATE$  ,&P$MIN
 000000                                    TRAN$  &K$REC

 000000                                   STATE$  ,&P$REC
 000000                                    TRAN$  &K$BUFS

 000000                                   STATE$  ,&P$RBM
 000000                                    TRAN$  $DNUMB, $EXIT,  .SYTHR
```

```
Symbol table

K$NODS= 000132          K$UBR = 000223          LISCOM  000216R      002 OBJO     011232R      002 OO.VFY= ****** GX
K$NONR= 000133          K$UBS = 000224          LISLOG  000410R      002 OC.APR= ****** GX         OPEVE   002510R      002
K$NUM = 000134          K$UBT = 000225          LISMOD  000470R      002 OC.CHN= ****** GX         OPEVE1  002532R      002
K$OBJ = 000135          K$UNT = 000226          LISOPT  000434R      002 OC.CST= ****** GX         OPLOG   002372R      002
K$OBJS= 000136          K$URM = 000227          LIST    000134R      002 OC.CTM= ****** GX         OPLOG1  002424R      002
K$OFF = 000137          K$USE = 000231          LIXAC   000666R      002 OC.DTE= ****** GX         OPLOG2  002442R      002
K$ON  = 000140          K$USG = 000230          LIXPR   000600R      002 OC.HTM= ****** GX         OPSTA   002460R      002
K$OUT = 000141          K$VAL = 000232          LIXSR   000670R      002 OC.MBL= ****** GX         OP.NCT= ****** GX
K$PAR = 000142          K$VEC = 000233          LOC     011770R      002 OC.MRC= ****** GX         OP.NLT= ****** GX
K$PAS = 000143          K$VER = 000234          LOC1    012000R      002 OC.MWI= ****** GX         OP.STA= ****** GX
K$PER = 000144          K$VRSN= 000235          LOC2    012010R      002 OC.MXR= ****** GX         OS.BYT= ****** GX
K$PHA = 000145          K$WIN = 000236          LOGID   002656R      002 OC.NUM= ****** GX         OS.CCB= ****** GX
K$POL = 000146          K$XCI = 000252          LOGOPT  002100R      002 OC.RPR= ****** GX         OS.LOC= ****** GX
K$PRI = 000147          K$XCNA= 000246          LSTATE  006320R      002 OC.RTM= ****** GX         OS.NAM= ****** GX
K$PRO = 000150          K$XDA = 000253          LXPRAB  000616R      002 OC.STA= ****** GX         OS.PAR= ****** GX
K$PROS= 000151          K$XDL = 000247          LXSRAB  000704R      002 OC.TRI= ****** GX         OS.RBN= ****** GX
K$PRT = 000152          K$XDLV= 000260          MASK    006566R      002 OC.USG= ****** GX         OS.RBS= ****** GX
K$PUR = 000153          K$XDMC= 000256          MBL     005666R      002 OE.ACO= ****** GX         OS.SBN= ****** GX
K$RCL = 000154          K$XDMF= 000272          MCIR    014044R      002 OE.ADD= ****** GX         OS.THR= ****** GX
K$REC = 000155          K$XDMP= 000261          MCIRD   014072R      002 OE.AHO= ****** GX         OVFY    011404R      002
K$RES = 000156          K$XDMR= 000273          MMDA    013612R      002 OE.AMA= ****** GX         PAR     011716R      002
K$ROU = 000162          K$XDMV= 000270          MODSPR  015016R      002 OE.HOS= ****** GX         PH4AD   004476R      002
K$ROUT= 000163          K$XDN = 000257          MODXAC  014636R      002 OE.ID  = ****** GX        PLX     006410R      002
K$RSTR= 000157          K$XDP = 000243          MODXPR  014606R      002 OE.LNK= ****** GX         PMXACA  002266R      002
K$RTR = 000160          K$XDPV= 000271          MODXSV  014622R      002 OE.MXC= ****** GX         PMXDST  002324R      002
K$RTRS= 000161          K$XDQ = 000251          MODX29  014652R      002 OE.MXH= ****** GX         PMXPRA  002170R      002
K$SEC = 000164          K$XDTE= 000262          MRC     005714R      002 OE.NAM= ****** GX         PMXPRX  002200R      002
K$SEG = 000165          K$XDU = 000245          MRE     013666R      002 OE.NCO= ****** GX         PMXPR1  002250R      002
K$SER = 000170          K$XDUP= 000255          MRT     007070R      002 OE.NN  = ****** GX        PMXSRA  002300R      002
K$SET = 000166          K$XDV = 000263          MSPR1   015026R      002 OE.RTM= ****** GX         PMXSRV  002310R      002
K$SHO = 000171          K$XDZ = 000264          MSPR2   015036R      002 OE.SUB= ****** GX         PNAM    011700R      002
K$SIN = 000172          K$XKDP= 000265          MWI     014770R      002 OE.VFY= ****** GX         PNDCT   001722R      002
K$SIZ = 000173          K$XKDZ= 000266          MWIL    007144R      002 OL.BAU= ****** GX         PNDIA   002052R      002
K$SMA = 000174          K$XKL = 000267          MXC     013640R      002 OL.BSZ= ****** GX         PNDMP   002064R      002
K$SNA = 000175          K$XKMY= 000274          MXR     013742R      002 OL.CSR= ****** GX         PNDPA   001706R      002
K$SPD = 000176          K$XPCL= 000254          MX25    014674R      002 OL.CTM= ****** GX         PNHAR   001756R      002
K$SRV = 000167          K$XQNA= 000250          MX25A   014704R      002 OL.DPR= ****** GX         PNHOS   001634R      002
K$STA = 000177          K$XUNA= 000244          MX25B   014720R      002 OL.HBT= ****** GX         PNLOD   002026R      002
K$STS = 000200          K$X25 = 000237          MX29    014732R      002 OL.PLX= ****** GX         PNLOD2  002036R      002
K$SUBA= 000202          K$X29 = 000240          MX29A   014742R      002 OL.PRI= ****** GX         PNLOD3  002046R      002
K$SUM = 000201          K$3   = 000241          MX29B   014756R      002 OL.RTC= ****** GX         PNOPT   001424R      002
K$SYS = 000203          K$4   = 000242          NEXT    006556R      002 OL.RTT= ****** GX         PNSCI   001736R      002
K$TER = 000204          K$9600= 000070          NNAM    014016R      002 OL.SEC= ****** GX         PNSDV   001640R      002
K$TIM = 000205          LCT     007014R     002 NODADD  004444R      002 OL.STA= ****** GX         PNSLO   002002R      002
K$TOP = 000206          LINCIR  007260R     002 NODFL   004522R      002 OL.URM= ****** GX         PNSNV   001660R      002
K$TRA = 000207          LINE    006046R     002 NODID   004374R      002 OL.VEC= ****** GX         PNSPA   001770R      002
K$TRIB= 000210          LINID   007256R     002 NODID1  004424R      002 OL.WSZ= ****** GX         PNSPA2  001776R      002
K$TYP = 000211          LINMAX  004706R     002 NODID2  004440R      002 OM.CON= ****** GX         PNTLO   002014R      002
K$UBE = 000212          LINOPT  006072R     002 NODID3  004416R      002 OM.EVE= ****** GX         POOL    011632R      002
K$UBF = 000213          LIN0    006060R     002 NODNAM  002764R      002 OM.FIL= ****** GX         PRI     006372R      002
K$UBH = 000214          LIN1    007310R     002 NODND   004510R      002 OM.KEV= ****** GX         PROC    004600R      002
K$UBJ = 000215          LIN2    007322R     002 NODN1   004466R      002 OM.MON= ****** GX         PROID   005002R      002
K$UBK = 000216          LIN3    007336R     002 NODN2   004530R      002 OM.STA= ****** GX         PROPT   004624R      002
K$UBL = 000217          LIN4    007350R     002 OBJI    014352R      002 ONAME    011314R      002 PROO    004612R      002
K$UBM = 000220          LIN5    007366R     002 OBJID   011440R      002 OO.COP= ****** GX         PRSKTB  000000RG     003
K$UBN = 000221          LISCM0  000176R     002 OBJN    014334R      002 OO.NAM= ****** GX         PRSSTB  000000RG     002
K$UBP = 000222          LISCM1  000210R     002 OBJOPT  011244R      002 OO.USR= ****** GX         PSTATE  004662R      002
```

```
    107                                                 .SBTTL  $CFE - CFE ROOT SEGMENT
    108                                         ;+
    109                                         ; *** - $CFE - CFE ROOT SEGMENT
    110                                         ;
    111                                         ; THE DECNET UTILITY CFE (CONFIGURATION FILE EDITOR) ALLOWS THE DECNET
    112                                         ; USER TO MODIFY PARTS OF THE NETWORK DATA BASE AS SYSTEM/NETWORK NEEDS
    113                                         ; CHANGE. THIS UTILITY MODIFIES THE FILE "CETAB.MAC", AND IS NOT PARTICULAR
    114                                         ; ABOUT THE TYPE OF SYSTEM THAT IT WAS GENERATED UNDER (I.E. YOU CAN MODIFY
    115                                         ; THE "CETAB" FILE FOR AN RSX-11S SYSTEM WHILE RUNNING CFE UNDER RSX-11M, ETC.)
    116                                         ; USERS INTENDING TO MODIFY THE FILE "CETAB.MAC" BY USING A STANDARD TEXT
    117                                         ; EDITOR, DO SO AT THEIR OWN RISK !!!
    118                                         ;
    119                                         ; INPUT:
    120                                         ;       NONE.
    121                                         ;
    122                                         ; OUTPUT:
    123                                         ;       THE FILE "CETAB.MAC" IS UPDATED.
    124                                         ;
    125                                         ;-
    126
    127                                                 .ENABL  LSB
    128 000034                                  $CFE::
    129 000034  004767  000000G                         CALL    $CFINI          ; INITIALIZE THE END OF TASK BUFFERS AND OPEN FILES
    130 000040  103464                                  BCS     30$             ; IF CS, FATAL ERROR
    131 000042  004767  000000G                 10$:    CALL    $CFCOM          ; ELSE, GET A PARSED COMMAND LINE
    132 000046  103403                                  BCS     $FIN            ; IF CS, EOF
    133 000050  004767  000000G                         CALL    $CFDSP          ; DISPATCH ON THE COMMAND CODE
    134 000054  000772                                  BR      10$             ; GET NEXT COMMAND
    135                                         ;
    136                                         ; NORMAL EXIT
    137                                         ;
    138 000056                                  $FIN::
    139 000056  004767  000000G                         CALL    $CFUPD          ; UPDATE THE "CETAB.MAC" FILE
    140 000062  103443                                  BCS     .EXIT           ; IF CS, ERROR - DELETE FILE
    141 000064  032767  000000G 000000G                 BIT     #SW.PUR,$SWICH  ; PURGE THE INPUT FILE ?
    142 000072  001404                                  BEQ     20$             ; IF EQ, NO
    143 000074                                          DELET$  #$IFDB          ; DELETE THE INPUT FILE
    144 000104  012700  000000G                 20$:    MOV     #$OFDB,R0       ; POINT TO OUTPUT FDB
    145 000110  016046  000102                          MOV     F.FNB+N.FID(R0),-(SP) ; SAVE FILE NUMBER FROM FDB
    146 000114                                          CLOSE$  #$OFDB          ; CLOSE THE OUTPUT FILE
    147 000124  012660  000102                          MOV     (SP)+,F.FNB+N.FID(R0) ; RESTORE FILE NUMBER IN FDB
    148 000130  105767  000000G                         TSTB    $FIATR          ; CHECK FUNCTION BYTE
    149 000134  001422                                  BEQ     25$             ; IF EQ, READ ATTRIBUTES FAILED
    150 000136  112767  000001  000000G                 MOVB    #1,$FIATR       ; ELSE SET DIRECTION FOR FILE ATTRIBUTES
    151 000144  116067  000042  177632                  MOVB    F.LUN(R0),WATDPB+Q.IOLU ; SET LUN IN DPB
    152 000152                                          DIR$    #WATDPB         ; SET OUTPUT FILE OWNER AND PROTECTION
    153 000160  000410                                  BR      25$             ; AND LEAVE
    154                                         ;
    155                                         ; CONSISTENCY ERROR
    156                                         ;
    157 000162                                  .CONER::
    158 000162                                          ERROR$  #ERRCON         ; PRINT ERROR MESSAGE
    159                                         ;
    160                                         ; ABNORMAL EXIT
    161                                         ;
    162 000172                                  .EXIT::
    163 000172                                          DELET$  #$OFDB          ; DELETE THE OUTPUT FILE
```

```
CFERQB - DYNAMIC BUFFER MANAGEM MACRO V05.03b Saturday 29-Jun-85 05:13  Page 6-3
Symbol table

$RLBF   000102RG        $SAVRG= ****** GX       $XXLEN= ****** GX       .EXINC= 000020 G        .RQCB   000206RG
$RQBF   000000RG        $XXBUF= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000346    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8312 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.18
SY:CFERQB.V2,[132,134]CFERQB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFERQB
```

```
CFESTB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 2

MACRO CROSS REFERENCE                                    CREF    04.00

MACRO NAME        REFERENCES

DBGTP$            #5-106    #5-112    #5-113    #5-114    #5-115    #5-116    #5-117    #5-118    #5-119    #5-120
                  #5-121    #5-122    #5-123    #5-124    #5-125    #5-126    #5-127    #5-128    #5-129    #5-130
                  #5-131    #5-132    #5-133    #5-134    #5-135    #5-136    #5-137    #5-138    #5-139    #5-140
                  #5-141    #5-142    #5-143    #5-144    #5-145    #5-146    #5-147    #5-148
EPRINT            #4-77      6-198
ERROR$            #4-66      6-186     8-240
ISTAT$            #4-91      5-106
MTRAN$            #5-106
PRINT             #4-61
STATE$            #4-91      5-110    #5-153
TRAN$             #4-91     #5-111    #5-112    #5-113    #5-114    #5-115    #5-116    #5-117    #5-118    #5-119
                  #5-120    #5-121    #5-122    #5-123    #5-124    #5-125    #5-126    #5-127    #5-128    #5-129
                  #5-130    #5-131    #5-132    #5-133    #5-134    #5-135    #5-136    #5-137    #5-138    #5-139
                  #5-140    #5-141    #5-142    #5-143    #5-144    #5-145    #5-146    #5-147    #5-148
```

```
**FILE**ID**CFESUB
```

```
                                        .SBTTL  $CAT5 - ASCII TO RAD50 CONVERSION
                                        .SBTTL  .CAT5 - ASCII TO EXTENDED RAD50 CONVERSION ("*")
                                        .SBTTL  .CAT5E- ASCII TO EXTENDED RAD50 CONVERSION ("-")
                                ;+
                                ; ** $CAT5 - ASCII TO RAD50 CONVERSION
                                ; ** .CAT5 - ASCII TO EXTENDED RAD50 CONVERSION ("*")
                                ; ** .CAT5E- ASCII TO EXTENDED RAD50 CONVERSION ("-")
                                ;
                                ; INPUTS        R0 = BUFFER ADDRESS
                                ;               R1 <> 0 TO ALLOW ".", = 0 TO FAIL ON "."
                                ;               .CAT5 ENTRY POINT ALLOWS "*"
                                ;
                                ; OUTPUT        R0 = BUFFER ADDRESS
                                ;               R2 = CHARACTER THAT STOPPED CONVERSION
                                ;-

000404     132     101     071  CNTRL:  .BYTE   'Z,'A,'9,'0,'$,'*,'.  ;CONTROL BYTE STRING
000407     060     044     052
000412     056
                                        .EVEN

000414  003100                  CVTBL:  .WORD   50*50
000416  000050                          .WORD   50
000420  000001                          .WORD   1

                                        .ENABL  LSB

000422  112767  000055  177761  .CAT5E::MOVB    #'-,CNTRL+5      ;EXTEND RAD50 WITH HYPHEN
000430  000403                          BR      1$               ;JOIN COMMON CODE

000432  112767  000052  177751  .CAT5:: MOVB    #'*,CNTRL+5      ;EXTEND RAD50 WITH ASTERISK
000440  005727                  1$:     TST     (PC)+            ;INDICATED EXTENDED MODE

000442  000261                  $CAT5:: SEC                      ;NO EXTENDED MODE
000444  004567  000000G                 JSR     R5,$SAVRG        ;SAVE REGISTERS
000450  006046                          ROR     -(SP)            ;SAVE EXTENDED MODE INDICATOR
000452  005701                          TST     R1               ;SAVE PERIOD DISPOSITION FLAG
000454  001001                          BNE     5$               ;BR IF "." ALLOWED
000456  000261                          SEC                      ;SET PERIOD FLAG
000460  106016                  5$:     RORB    (SP)             ;SAVE PERIOD FLAG
000462  005001                          CLR     R1               ;CLEAR ACCUMULATOR
000464  012703  000414'                 MOV     #CVTBL,R3        ;GET CONVERSION TABLE
000470  112005                  10$:    MOVB    (R0)+,R5         ;GET NEXT CHARACTER
000472  012704  000404'                 MOV     #CNTRL,R4        ;GET CONTROL STRING
000476  122405                          CMPB    (R4)+,R5         ;RAD50 CHARACTER?
000500  103450                          BLO     60$              ;NO
000502  122405                          CMPB    (R4)+,R5         ;ALPHABETIC?
000504  101425                          BLOS    50$              ;YES
000506  122405                          CMPB    (R4)+,R5         ;RAD50 CHARACTER?
000510  103444                          BLO     60$              ;NO
000512  122405                          CMPB    (R4)+,R5         ;NUMERIC?
000514  101417                          BLOS    40$              ;YES
000516  122405                          CMPB    (R4)+,R5         ;"$"?
000520  001413                          BEQ     30$              ;YES
000522  122405                          CMPB    (R4)+,R5         ;"*" or "-" ?
000524  001405                          BEQ     15$              ;YES
000526  105716                          TSTB    (SP)             ;PERIOD A RAD50 CHARACTER?
```

```
                                                                  H 16
CFESUB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14     PAGE 3

SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

R$LEN      000024            5-99
R$LENX   = 000024   G       #5-99
R$$EIS   = ******            12-405
R.ADD      000010           #5-94
R.FLAG     000012           #5-94
R.LEN      000014           #5-94          5-94
R.LENX   = 000014   G       #5-94
R.LNK      000000           #5-94
R.NAM      000002           #5-94
SF.CIR   = 000010           #5-103
SF DEV   = 000020           #5-103
SF.DIA   = 004000           #5-103
SF.DPA   = 000100           #5-103
SF.DPC   = 000200           #5-103
SF.DUM   = 010000           #5-103
SF.HAD   = 000004           #5-103
SF.HST   = 000002           #5-103
SF.LOA   = 002000           #5-103
SF.PH3   = 020000           #5-103
SF.PSS   = 000040           #5-103
SF.SEC   = 000400           #5-103
SF.TER   = 001000           #5-103
S$LEN      000032            5-102
S$LENX   = 000034   G       #5-102
S.ADD      000004           #5-103
S.CIR      000016           #5-103
S.DEV      000020           #5-103
S.DIA      000046           #5-103
S.DPA      000032           #5-103
S.DPC      000036           #5-103
S.DUM      000050           #5-103
S.FLAG     000002           #5-103
S.HAD      000010           #5-103
S.HST      000006           #5-103
S.LENG     000052           #5-103         5-103
S.LENX   = 000054   G       #5-103
S.LNK      000000           #5-103
S.LOA      000044           #5-103
S.PSS      000022           #5-103
S.SEC      000040           #5-103
S.TER      000042           #5-103
T$FLAG     000044           #4-63
T$LIF      000013           #4-63
T$LIFL     000013           #4-63
T$LIFO     000013           #4-63
T$LIFS     000013           #4-63
T$LIN      000000           #4-63
T$LIPS     000006           #4-63
T$LLD      000012           #4-63
T$LLDC     000045           #4-63
T$LLDL     000012           #4-63
T$LLDO     000012           #4-63
```

```
                                                                  I 16
CFESUB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14     PAGE 4
```

```
        000010                          E2.INA  == 10            ; INACTIVITY TIMER
        000020                          E2.INC  == 20            ; INCOMING TIMER
        000040                          E2.OUT  == 40            ; OUTGOING TIMER
        000100                          E2.RET  == 100           ; RETRANSMIT FACTOR


        000610                            OE.NSP == OE.LNK!OE.VFY!OE.NCO       ; NSP PARAMETERS
        000174                            E2.NSP == E2.SEG!E2.INA!E2.INC!E2.OUT!E2.RET ; DITTO
        001007                            OE.NOD == OE.ID!OE.NAM!OE.ADD!OE.HOS ; NODE PARAMETERS
        070000                            OE.LV2 == OE.AMAX!OE.AHOP!OE.ACOS    ; LEVEL 2 ROUTING PARAMETERS
        072160                            OE.ROU == OE.NN!OE.MXC!OE.MXH!OE.RTM!OE.LV2 ; ROUTING PARAMETERS
        000003                            E2.ROU == E2.BEA!E2.BRT              ; LIKEWISE
                                ;
                                ; SYSTEM OPTIONS
                                ;
        000001                          OS.LOC == 1              ; =1 IF LOADING LOCATION PRESENT
        000002                          OS.NAM == 2              ; =1 IF DYNAMIC PARTITION NAME PRESENT
        000004                          OS.PAR == 4              ; =1 IF PARTITION NAME IS PRESENT
        000010                          OS.CCB == 10             ; =1 IF CCB NUMBER PRESENT
        000020                          OS.RBN == 20             ; =1 IF RDB NUMBER PRESENT
        000040                          OS.RBS == 40             ; =1 IF RDB SIZE PRESENT
        000100                          OS.SBN == 100            ; =1 IF SDB NUMBER PRESENT
        000200                          OS.THR == 200            ; =1 IF RDB THRESHOLD VALUE IS PRESENT
        000400                          OS.BYT == 400            ; =1 IF POOL BYTE-AREA PRESENT
        001000                          OS.BLK == 1000           ; =1 IF POOL BLOCK-AREA PRESENT
        001407                            OS.PRT == OS.PAR!OS.NAM!OS.BLK!OS.BYT!OS.LOC ; PARTITION PARAMETERS
        000370                            OS.BUF == OS.CCB!OS.RBN!OS.RBS!OS.SBN!OS.THR ; BUFFER PARAMETERS
                                ;
                                ; NODE OPTIONS
                                ;
        000001                          ON.NAM == 1              ; =1 IF NODE NAME PRESENT
        000001                          ON.ADD == 1              ; =1 IF NODE ADDRESS PRESENT
                                ;
                                ; REMOTE NODE OPTIONS
                                ;
        000002                          RN.NAM  ==2              ;Remote name present
        000004                          RN.HOS  == 4             ;Host node address
        000010                          RN.SDV  == 10            ;Service device
        000020                          RN.SNV  == 20            ;Service node version
        000040                          RN.DPA  == 40            ;Dump address
        000100                          RN.DCT  == 100           ;Dump count
        000200                          RN.SCI  == 200           ;Service circuit
        000400                          RN.HAR  == 400           ;Hardware address
        001000                          RN.SPA  == 1000          ;Service password
        002000                          RN.SLO  == 2000          ;Secondary loader file
        004000                          RN.TLO  == 4000          ;Tertiary loader file
        010000                          RN.LOD  == 10000         ;Load file
        020000                          RN.DIA  == 20000         ;Diagnostics file
        040000                          RN.DMP  == 40000         ;Dump file
                                ;
                                ; ALL SER$DF/FIL$DF REMOTE NODE PARAMETERS
                                ;
        001774                          $$$      =  RN.HOS!RN.SDV!RN.SNV!RN.DPA!RN.DCT!RN.SCI!RN.HAR!RN.SPA
        077774                          RN.XXX  == $$$!RN.SLO!RN.TLO!RN.LOD!RN.DIA!RN.DMP



                                ;
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

CS.EVT   = 000017  G        #12-1107
CS.FEA   = 000015  G        #12-1105
CS.FIL   = 000045  G        #12-1140
CS.FLG   = 010000  G        #12-1152
CS.LLC   = 000003  G        #12-1095
CS.LOG   = 000020  G        #12-1108
CS.MOD   = 100000  G        #12-1149
CS.MSK   = 000400  G        #12-1178
CS.NET   = 000400  G        #12-1161
CS.NOD   = 000012  G        #12-1102
CS.NUM   = 000400  G        #12-1173
CS.OBJ   = 000014  G        #12-1104
CS.PAR   = 000011  G        #12-1101
CS.PSN   = 000022  G        #12-1113
CS.PVC   = 000030  G        #12-1119
CS.RDT   = 000035  G        #12-1124
CS.REM   = 000013  G        #12-1103
CS.ROU   = 000016  G        #12-1106
CS.SAC   = 000041  G        #12-1132
CS.SAD   = 000042  G        #12-1133
CS.SER   = 000044  G        #12-1139
CS.SHI   = 001000  G        #12-1174
CS.SLO   = 002000  G        #12-1175
CS.SLT   = 000002  G        #12-1094
CS.SNA   = 000040  G        #12-1131
CS.SPU   = 000043  G        #12-1134
CS.STA   = 000010  G        #12-1100
CS.SVC   = 000021  G        #12-1112
CS.THI   = 000400  G        #12-1158
CS.TLO   = 001000  G        #12-1159
CS.TOP   = 000400  G        #12-1156
CS.TRB   = 020000  G        #12-1151
CS.TSK   = 000400  G        #12-1171
CS.UNI   = 040000  G        #12-1150
CS.UNT   = 000007  G        #12-1099
CS.VAL   = 001000  G        #12-1179
CS.X2P   = 000024  G        #12-1115
CS.X29   = 000025  G        #12-1116
CS.X3P   = 000023  G        #12-1114
CU$BIL   = 000001  G        #12-1169
C$HHI      000006  G        #9-535
C$HLEN   = 000010  G        #9-536
C$HLO      000004  G        #9-534
C$NCNT     000004  G        #9-389
C$NCSR     000010  G        #9-391
C$NLEN   = 000016  G        #9-394
C$NPRI     000012  G        #9-392
C$NURM     000014  G        #9-393
C$NVEC     000006  G        #9-390
C$UCUG     000010  G        #9-548
C$UFLG     000012  G        #9-549
C$ULEN   = 000013  G        #9-550
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 15
SYMBOL CROSS REFERENCE                                   CREF    04.00
SYMBOL   VALUE                REFERENCES

$SDAT    = 001550 RG      #7-295
$SDBNM     001526 RG      #7-176
$SLUCT   = 001622 RG      #7-290       7-291
$SLUS    = 001624 RG      #7-291
$SMOD    = 001754 RG      #7-283       7-284
$SNAPT     001426 RG      #7-134
$SNLU    = 001744 RG      #7-301       7-302
$SNSID   = 001746 RG      #7-302       7-303
$SPARE   = 001734 RG      #7-300       7-301
$SPLU    = 001744 RG      #7-282       7-283
$SPSW    = 001774 RG      #7-285
$SPUNM   = 001724 RG      #7-299       7-300
$SSERN   = 001750 RG      #7-303       7-304
$SSSCP   = 001754 RG      #7-304
$STADR   = 001566 RG      #7-226
$STAN      001540 RG      #7-185       7-252
$SUB1      002004 RG      #7-264
$SUB2      002006 RG      #7-265
$SUID    = 001764 RG      #7-284       7-285
$SVUPD     001430 RG      #7-135
$SWICH     001460 RG      #7-146
$THRSH     001530 RG      #7-177
$TIDPB     000354 RG      #7-127
$TMLST     001420 RG      #7-132
$TNAM      001576 RG      #7-203       7-248       7-249
$TNUM      001602 RG      #7-204       7-244       7-251
$UMRIN     001512 RG      #7-167
$UMRUP     001514 RG      #7-168
$UNTNM     001536 RG      #7-184
$URM     = 001562 RG      #7-227
$VEC       001550 RG      #7-191       7-219       7-225       7-254       7-258       7-295       7-309
$VLVL      001620 RG      #7-208       7-224       7-239       7-255
$WILD      001462 RG      #7-147
$WSZ     = 001556 RG      #7-246
$$$      = 001774        #11-914      11-915
$$$ARG   = 000003         #6-95        6-95       #6-95        6-95       #6-95        6-95       #6-95        6-95        6-95
                         #7-127       7-127      #7-127       7-127      #7-127       7-127      #7-127       7-127       7-127
                         #7-128       7-128      #7-128       7-128      #7-128       7-128      #7-128       7-128       7-128
$$$GLB   = ******          6-95        6-95        6-95        6-95        6-95       7-127       7-128
$$$T1    = 000002         #6-95
$$$T2    = 000166 R       #6-95        6-95
.FSRCB   = ******   G      6-92
 RNFSP     001512 RG      #7-166
...PC1   = 000142 R       #6-95        6-95        6-95        6-95        6-95        6-95       #6-95        6-95        6-95
                           6-95        6-95        6-95       #6-95        6-95        6-95      #6-104       6-105       6-105
                          6-106       6-106       6-106      #6-110       6-111       6-111       6-112       6-112       6-113
                          6-113       6-113       6-115       6-115      #6-115       6-115       6-115       6-115       6-115
                          6-115      #6-115
...PC2   = 000336 R       #6-95        6-95       #6-95        6-95       #6-95        6-95       #6-95        6-95        6-95
                          #6-95        6-95        6-95       #6-95        6-95       #6-95        6-95       #6-95        6-95
                         #6-105       6-105      #6-105       6-105      #6-106       6-106      #6-106       6-106      #6-106
                          6-106      #6-111       6-111      #6-111       6-111      #6-112       6-112      #6-112       6-112
```

```
                .TITLE  CFELIS - PERFORM LIST COMMAND
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - PERFORM LIST COMMAND
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  14-DEC-79
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;
```

```
Symbol table

BUFSZ = 000122          FX.LEN= 000000          F.FNAM= 000110          F.VBSZ= 000060          S.FATT= 000016
CSBLK   000300R         F.ACTL= 000076          F.FNB = 000102          IE.BNM= ****** GX       S.FDB = 000140
CS.DIF= 000002          F.ALOC= 000040          F.FTYP= 000116          IOSB    000030R         S.FNAM= 000006
CS.DVF= 000004          F.BBFS= 000062          F.FVER= 000120          IO.RAT= ****** GX       S.FNB = 000036
CS.EQU= 000040          F.BDB = 000070          F.HIBK= 000004          N.ANM1= 000002          S.FNBW= 000017
CS.INP= 000001          F.BGBC= 000057          F.LUN = 000042          N.ANM2= 000024          S.FNTY= 000004
CS.MOR= 000020          F.BKDN= 000026          F.MBCT= 000054          N.DID = 000024          S.FTYP= 000002
CS.NMF= 000001          F.BKDS= 000020          F.MBC1= 000055          N.DVNM= 000032          S.NFEN= 000020
CS.OUT= 000002          F.BKEF= 000050          F.MBFG= 000056          N.FID = 000000          $BUFI   000156RG
CS.WLD= 000010          F.BKP1= 000051          F.NRBD= 000024          N.FNAM= 000006          $BUFR   000034RG
C.CMLD= 000002          F.BKST= 000024          F.NREC= 000030          N.FTYP= 000014          $FIATR= ****** GX
C.DEVD= 000006          F.BKVB= 000064          F.OVBS= 000030          N.FVER= 000016          $IFDB = ****** GX
C.DIRD= 000012          F.CHR = 000075          F.RACC= 000016          N.NEXT= 000022          $IOPEN  000354RG
C.DSDS= 000006          F.CNTG= 000034          F.RATT= 000001          N.STAT= 000020          $OFDB = ****** GX
C.FILD= 000016          F.DFNB= 000046          F.RCNM= 000034          N.UNIT= 000034          $OPEN   000530RG
C.MKW1= 000024          F.DSPT= 000044          F.RCTL= 000017          Q.IOAE= 000012          $RDATA  000476RG
C.MKW2= 000026          F.DVNM= 000134          F.RSIZ= 000002          Q.IOEF= 000006          $WDATA  000662RG
C.SIZE= 000054          F.EFBK= 000010          F.RTYP= 000000          Q.IOFN= 000002          $$$ARG= 000002
C.STAT= 000001          F.EFN = 000050          F.SEQN= 000100          Q.IOLU= 000004          $$$OST= 000014
C.SWAD= 000022          F.EOBB= 000032          F.SPDV= 000072          Q.IOPL= 000014          .CSI1 = ****** G
C.TYPR= 000000          F.ERR = 000052          F.SPUN= 000074          Q.IOPR= 000007          .CSI2 = ****** G
FO.RD = ****** GX       F.EXT = 000072          F.STBK= 000036          Q.IOSB= 000010          .GET  = ****** G
FO.WRT= ****** GX       F.FACC= 000043          F.UNIT= 000136          RATDPB  000000R         .OPEN = ****** G
FX.DIR= 000004          F.FFBY= 000014          F.URBD= 000020          S.ANM1= 000014          .PUT  = ****** G
FX.DLN= 000003          F.FLG = 000074          F.VBN = 000064          S.ANM2= 000005          ...TPC= 000006
FX.DMX= 000002

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000674    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10008 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:15.31
SY:CFEOPN.V2,[132,134]CFEOPN/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEOPN
```

```
                                         .SBTTL  PURLIN  - PURGE LINE
                                ;+
                                ; ** PURLIN - PURGE LINE
                                ;
                                ; INPUTS:        $DVNAM = PARSED LINE ID
                                ;                $WILD = PARSED WILDCARD FLAGS
                                ;
                                ; OUTPUT:        COUNTER TIMER PURGED.
                                ;-

 000734  016705  000000G        PURLIN: MOV      $TMLST,R5        ; POINT TO START OF TEMPLATES
 000740  004767  000000G        10$:    CALL     .FNSLT           ; NEXT SLT
 000744  103403                         BCS      20$              ; IF CS, NO MORE SLTS
 000746  005065  000000G                CLR      S$LCTT(R5)       ; PURGE COUNTER TIMER
 000752  000772                         BR       10$              ; CONTINUE
 000754  000207                 20$:    RETURN
```

```
    168 000000                                      KEYWD$  ''LOOP'',          K$LOO
    169 000000                                      KEYWD$  ''MASK'',          K$MSK
    170 000000                                      KEYWD$  ''MAXIMUM'',       K$MAX
    171 000000                                      KEYWD$  ''MINIMUM'',       K$MIN
    172 000000                                      KEYWD$  ''MODULE'',        K$MOD
    173 000000                                      KEYWD$  ''MONITOR'',       K$MON
    174 000000                                      KEYWD$  ''MULTIPOINT'',    K$MUL
    175 000000                                      KEYWD$  ''NAME'',          K$NAM
    176 000000                                      KEYWD$  ''NETWORK'',       K$NET
    177 000000                                      KEYWD$  ''NODE'',          K$NOD
    178 000000                                      KEYWD$  ''NODES'',         K$NODS
    179 000000                                      KEYWD$  ''NONROUTERS'',    K$NONR
    180 000000                                      KEYWD$  ''NUMBER'',        K$NUM
    181 000000                                      KEYWD$  ''OBJECT'',        K$OBJ
    182 000000                                      KEYWD$  ''OBJECTS'',       K$OBJS
    183 000000                                      KEYWD$  ''OFF'',           K$OFF
    184 000000                                      KEYWD$  ''ON'',            K$ON
    185 000000                                      KEYWD$  ''OUTGOING'',      K$OUT
    186 000000                                      KEYWD$  ''PARTITION'',     K$PAR
    187 000000                                      KEYWD$  ''PASSWORD'',      K$PAS
    188 000000                                      KEYWD$  ''PERMANENT'',     K$PER
    189 000000                                      KEYWD$  ''PHASE'',         K$PHA
    190 000000                                      KEYWD$  ''POOL'',          K$POL
    191 000000                                      KEYWD$  ''PRIORITY'',      K$PRI
    192 000000                                      KEYWD$  ''PROCESS'',       K$PRO
    193 000000                                      KEYWD$  ''PROCESSES'',     K$PROS
    194 000000                                      KEYWD$  ''PROTOCOL'',      K$PRT
    195 000000                                      KEYWD$  ''PURGE'',         K$PUR
    196 000000                                      KEYWD$  ''RECALL'',        K$RCL
    197 000000                                      KEYWD$  ''RECEIVE'',       K$REC
    198 000000                                      KEYWD$  ''RESETS'',        K$RES
    199 000000                                      KEYWD$  ''RESTARTS'',      K$RSTR
    200 000000                                      KEYWD$  ''RETRANSMIT'',    K$RTR
    201 000000                                      KEYWD$  ''RETRANSMITS'',   K$RTRS
    202 000000                                      KEYWD$  ''ROUTING'',       K$ROU
    203 000000                                      KEYWD$  ''ROUTERS'',       K$ROUT
    204 000000                                      KEYWD$  ''SECONDARY''      K$SEC
    205 000000                                      KEYWD$  ''SEGMENT'',       K$SEG
    206 000000                                      KEYWD$  ''SET'',           K$SET
    207 000000                                      KEYWD$  ''SERVER'',        K$SRV
    208 000000                                      KEYWD$  ''SERVICE'',       K$SER
    209 000000                                      KEYWD$  ''SHOW'',          K$SHO
    210 000000                                      KEYWD$  ''SINGLE'',        K$SIN
    211 000000                                      KEYWD$  ''SIZE'',          K$SIZ
    212 000000                                      KEYWD$  ''SMALL'',         K$SMA
    213 000000                                      KEYWD$  ''SNA'',           K$SNA
    214 000000                                      KEYWD$  ''SPEED'',         K$SPD
    215 000000                                      KEYWD$  ''STATE'',         K$STA
    216 000000                                      KEYWD$  ''STATUS'',        K$STS
    217 000000                                      KEYWD$  ''SUMMARY'',       K$SUM
    218 000000                                      KEYWD$  ''SUBADDRESSES'',  K$SUBA
    219 000000                                      KEYWD$  ''SYSTEM'',        K$SYS
    220 000000                                      KEYWD$  ''TERTIARY'',      K$TER
    221 000000                                      KEYWD$  ''TIMER'',         K$TIM
    222 000000                                      KEYWD$  ''TOPDOWN'',       K$TOP
    223 000000                                      KEYWD$  ''TRANSMIT'',      K$TRA
    224 000000                                      KEYWD$  ''TRIBUTARY'',     K$TRIB
```

DEFINE NODE COMMAND

```
    815 000000                                  TRAN$   !RNSDV3,$EXIT,   .RNSDV
    816 000000                                  TRAN$   $LAMDA, $FAIL,   $CLERR, ERRIPV, $ERROR
    817
    818 000000                                 STATE$   RNSDV3
    819 000000                                  TRAN$   &K$XDP, $EXIT
    820 000000                                  TRAN$   &K$XUNA,$EXIT
    821 000000                                  TRAN$   &K$XDU, $EXIT
    822 000000                                  TRAN$   &K$XCNA,$EXIT
    823 000000                                  TRAN$   &K$XDL, $EXIT
    824 000000                                  TRAN$   &K$XQNA,$EXIT
    825 000000                                  TRAN$   &K$XDQ, $EXIT
    826 000000                                  TRAN$   &K$XCI, $EXIT
    827 000000                                  TRAN$   &K$XDA, $EXIT
    828 000000                                  TRAN$   &K$XPCL,$EXIT
    829 000000                                  TRAN$   &K$XDUP,$EXIT
    830 000000                                  TRAN$   &K$XDMC,$EXIT
    831 000000                                  TRAN$   &K$XDN, $EXIT
    832 000000                                  TRAN$   &K$XDLV,$EXIT
    833 000000                                  TRAN$   &K$XDMP,$EXIT
    834 000000                                  TRAN$   &K$XDTE,$EXIT
    835 000000                                  TRAN$   &K$XDV, $EXIT
    836 000000                                  TRAN$   &K$XDZ, $EXIT
    837 000000                                  TRAN$   &K$XKDP,$EXIT
    838 000000                                  TRAN$   &K$XKDZ,$EXIT
    839 000000                                  TRAN$   &K$XKL, $EXIT
    340 000000                                  TRAN$   &K$XDMV,$EXIT
    841 000000                                  TRAN$   &K$XDPV,$EXIT
    842 000000                                  TRAN$   &K$XDMF,$EXIT
    843 000000                                  TRAN$   &K$XDMR,$EXIT
    844 000000                                  TRAN$   &K$XKMY,$EXIT
    845                                 ;
    846                                 ; Service node version
    847                                 ;
    848 000000                                 STATE$   RNSNV
    849 000000                                  TRAN$   &K$VRSN,RNSNV3
    850 000000                                  TRAN$   &K$SER
    851
    852 000000                                 STATE$   ,&P$SERO
    853 000000                                  TRAN$   &K$NOD,RNSNV1
    854 000000                                  TRAN$   $LAMDA,RNSNV2
    855
    856 000000                                 STATE$   RNSNV1
    857 000000                                  TRAN$   &K$VRSN,RNSNV3
    858 000000                                  TRAN$   $LAMDA,RNSNV3
    859
    860 000000                                 STATE$   RNSNV2
    861 000000                                  TRAN$   &K$VRSN
    862
    863 000000                                 STATE$   RNSNV3,&P$SERV
    864 000000                                  TRAN$   &K$PHA,RNSNV4
    865 000000                                  TRAN$   $LAMDA
    866 000000                                 STATE$   RNSNV4
    867                 000400                  SF.PH3 = 400
    868 000000                                  TRAN$   &K$3,    $EXIT,,        SF.PH3, $SWICH
    869 000000                                  TRAN$   <'3>,    $EXIT,,        SF.PH3, $SWICH
    870 000000                                  TRAN$   &K$4,    $EXIT
    871 000000                                  TRAN$   <'4>,    $EXIT
```

```
   1493 000000                          TRAN$   $DNUMB,$EXIT,.LIRTC
   1494                         ;
   1495                         ; RETRANSMIT TIMER
   1496                         ;
   1497 000000                          STATE$  RTT
   1498 000000                           TRAN$  &K$RTR
   1499 000000                          STATE$  ,&P$RTO
   1500 000000                           TRAN$  &K$TIM
   1501 000000                          STATE$  ,&P$RTT
   1502 000000                           TRAN$  $DNUMB,$EXIT,.LIRTT
   1503                         ;
   1504                         ; MAXIMUM WINDOW (FOR LINE)
   1505                         ;
   1506 000000                          STATE$  MWIL
   1507 000000                           TRAN$  &K$MAX
   1508 000000                          STATE$  ,           &P$MAX
   1509 000000                           TRAN$  &K$WIN
   1510 000000                          STATE$  ,           &P$MWI
   1511 000000                           TRAN$  $DNUMB, $EXIT,  .LIMWI
   1512
   1513                                 .SBTTL  SUB-EXPRESSIONS FOR CIRCUIT/LINE ID
   1514                         ;
   1515                         ; LOOK FOR CIRCUIT ID - REJECT IF NO SUCH CIRCUIT
   1516                         ;
   1517 000000                          STATE$  CIRID
   1518 000000                           TRAN$  &K$CIR,,,1,$CIRCT
   1519
   1520 000000                          STATE$  ,&P$CRID
   1521 000000                           TRAN$  !LINCIR,$EXIT,.CKDLM             ; TRANSPORT CIRCUIT
   1522 000000                           TRAN$  $STRNG,$EXIT,.CIRID              ; PSI CIRCUIT
   1523                         ;
   1524                         ; LOOK FOR CIRCUIT ID - ALLOW UNKNOWN PSI CIRCUITS
   1525                         ;
   1526 000000                          STATE$  CIRI1
   1527 000000                           TRAN$  &K$CIR,,,1,$CIRCT
   1528
   1529 000000                          STATE$  ,&P$CRID
   1530 000000                           TRAN$  !LINCIR,$EXIT,.CKDLM             ; TRANSPORT CIRCUIT
   1531 000000                           TRAN$  $STRNG,$EXIT,.CIRI1              ; PSI CIRCUIT
   1532
   1533                         ;
   1534                         ; LOOK FOR LINE ID
   1535                         ;
   1536 000000                          STATE$  LINID
   1537 000000                           TRAN$  &K$LIN
   1538
   1539                         ;
   1540                         ; COMMON PART OF LINE AND CIRCUIT
   1541                         ;
   1542 000000                          STATE$  LINCIR,&P$LNID
   1543 000000                           TRAN$  $RAD50,,.LINAM
   1544
   1545 000000                          STATE$
   1546 000000                           TRAN$  <'->
   1547
   1548 000000                          STATE$
   1549 000000                           TRAN$  $DNUMB,LIN1,.LICNT
   2137                                                  .SBTTL  DEFINE MODULE COMMAND
   2138                                           ;
   2139                                           ; DEFINE MODULE COMMAND
   2140                                           ;
   2141 000000                                           STATE$  DEFMOD
   2142 000000                                            TRAN$  !MODXPR,DEFXPR   ,,CO.MXP,$COMP
   2143 000000                                            TRAN$  !DXPRAB,$EXIT    ,,CO.MXP,$COMP
   2144 000000                                            TRAN$  !MODXSV,DEFXSV   ,,CO.MXS,$COMP
   2145 000000                                            TRAN$  !MODXAC,DEFXAC   ,,CO.MAC,$COMP
   2146 000000                                            TRAN$  !MODX29,DEFX29   ,,CO.MX9,$COMP
   2147 000000                                            TRAN$  !MODSPR,$EXIT,$SNDEF,CO.MSP,$COMP ; DEFINE MOD SNA-PROTOCOL
   2148                                           ;
   2149                                           ; DEFINE MODULE X25-PROTOCOL COMMAND
   2150                                           ;
   2151 000000                                           STATE$  DEFXPR, &P$DMOD
   2152 000000                                            TRAN$  !DXPRCH,$EXIT                   ; CHARACTERISTICS PARAMETERS
   2153 000000                                            TRAN$  $LAMDA
   2154 000000                                           STATE$  DXPRAB
   2155 000000                                            TRAN$  !DXPDDT,$EXIT                   ; DTE PARAMETERS
   2156 000000                                            TRAN$  !DXPCUG,$EXIT                   ; GROUP PARAMETERS
   2157                                           ;
   2158                                           ; DEFINE MODULE X25-SERVER CHARACTERISTICS
   2159                                           ;
   2160 000000                                           STATE$  DEFXSV, &P$DMOD
   2161 000000                                            TRAN$  !DFXSV2,DFXSV1
   2162 000000                                            TRAN$  !DSTLIS,$EXIT
   2163 000000                                           STATE$  DFXSV1
   2164 000000                                            TRAN$  $EOS,    $EXIT
   2165 000000                                            TRAN$  !DFXSV2,DFXSV1
   2166 000000                                           STATE$  DFXSV2
   2167 000000                                            TRAN$  !CT,     $EXIT,, XS.CTM,$OPRND
   2168 000000                                            TRAN$  !MCIR,   $EXIT,, XS.MXC,$OPRND
   2169                                           ;
   2170                                           ; DEFINE MODULE X25-ACCESS
   2171                                           ;
   2172 000000                                           STATE$  DEFXAC, &P$DMOD
   2173 000000                                            TRAN$  !DFXAC2
   2174 000000                                           STATE$  DFXAC1
   2175 000000                                            TRAN$  $EOS,EXIT
   2176 000000                                            TRAN$  !DFXAC2,DFXAC1
   2177 000000                                           STATE$  DFXAC2
   2178 000000                                            TRAN$  !DNUM,   $EXIT,,XA.NUM,$OPRND
   2179 000000                                            TRAN$  !DST,    $EXIT,,XA.DST,$OPRND
   2180                                           ;
   2181                                           ; DEFINE MODULE X29-SERVER CHARACTERISTICS
   2182                                           ;
   2183 000000                                           STATE$  DEFX29, &P$DMOD
   2184 000000                                            TRAN$  !DFX29B,DFX29A
   2185 000000                                            TRAN$  !DSTLIS,$EXIT
   2186 000000                                           STATE$  DFX29A
   2187 000000                                            TRAN$  $EOS,    $EXIT
   2188 000000                                            TRAN$  !DFX29B,DFX29A
   2189 000000                                           STATE$  DFX29B
   2190 000000                                            TRAN$  !CT,     $EXIT,, XS.CTM,$OPRND
   2191 000000                                            TRAN$  !MCIR,   $EXIT,, XS.MXC,$OPRND
   2192                                           ;
   2193                                           ; PARSE X25-SERVER/X29-SERVER DESTINATION
```

```
Symbol table

PURCIR  002130R     002 P$EXE = ****** GX       P$PUOP= ****** GX       RNLOD4  004232R     002 THRESH  012214R     002
PURCT   002154R     002 P$GRPO= ****** GX       P$PUR = ****** GX       RNOPT   003006R     002 TRIB    005530R     002
PURGE   001124R     002 P$HADR= ****** GX       P$RBM = ****** GX       RNSCI   003726R     002 UNCSR   006516R     002
PURLIN  002112R     002 P$HBO = ****** GX       P$RCO = ****** GX       RNSCI2  003744R     002 URM     006542R     002
PURNOD  001404R     002 P$HBT = ****** GX       P$REC = ****** GX       RNSDV   003274R     002 USER    011360R     002
PUROPT  001374R     002 P$HOP = ****** GX       P$RESO= ****** GX       RNSDV2  003312R     002 USG     005770R     002
P$ACOS= ****** GX       P$HOS = ****** GX       P$REST= ****** GX       RNSDV3  003342R     002 VEC     006354R     002
P$ACT = ****** GX       P$HTM = ****** GX       P$RETF= ****** GX       RNSLO   004120R     002 VFY1    010272R     002
P$ADD = ****** GX       P$HTO = ****** GX       P$ROU = ****** GX       RNSLO2  004130R     002 WC.CIR= ****** GX
P$AHOP= ****** GX       P$INA = ****** GX       P$ROUT= ****** GX       RNSNV   003512R     002 WC.CNT= ****** GX
P$AOP = ****** GX       P$INC = ****** GX       P$RPR = ****** GX       RNSNV1  003536R     002 WC.EVE= ****** GX
P$BEA = ****** GX       P$LAR = ****** GX       P$RSTO= ****** GX       RNSNV2  003546R     002 WC.LIN= ****** GX
P$BROO= ****** GX       P$LBN = ****** GX       P$RSTT= ****** GX       RNSNV3  003550R     002 WC.LOG= ****** GX
P$BRT = ****** GX       P$LBS = ****** GX       P$RTC = ****** GX       RNSNV4  003564R     002 WC.NOD= ****** GX
P$BUF = ****** GX       P$LIS = ****** GX       P$RTM = ****** GX       RNSPA   004062R     002 WC.OBJ= ****** GX
P$BYT = ****** GX       P$LIST= ****** GX       P$RTO = ****** GX       RNSPA2  004070R     002 WC.PRO= ****** GX
P$CALL= ****** GX       P$LNID= ****** GX       P$RTT = ****** GX       RNTLO   004154R     002 WC.TRI= ****** GX
P$CALT= ****** GX       P$LOC = ****** GX       P$SBN = ****** GX       RNTLO2  004164R     002 WC.UNT= ****** GX
P$CBN = ****** GX       P$LOF = ****** GX       P$SDV = ****** GX       RN.DCT= ****** GX           XA.DST= ****** GX
P$CFE = ****** GX       P$LOG = ****** GX       P$SEG = ****** GX       RN.DIA= ****** GX           XA.NUM= ****** GX
P$CHN = ****** GX       P$LPRI= ****** GX       P$SER = ****** GX       RN.DMP= ****** GX           XP.BIL= ****** GX
P$CHNL= ****** GX       P$MAX = ****** GX       P$SERC= ****** GX       RN.DPA= ****** GX           XP.CHN= ****** GX
P$CIST= ****** GX       P$MBD = ****** GX       P$SERO= ****** GX       RN.HAR= ****** GX           XP.CLT= ****** GX
P$CLM = ****** GX       P$MCIR= ****** GX       P$SERP= ****** GX       RN.HOS= ****** GX           XP.COU= ****** GX
P$CLRO= ****** GX       P$MCLR= ****** GX       P$SERV= ****** GX       RN.LOD= ****** GX           XP.CTM= ****** GX
P$CLRT= ****** GX       P$MIN = ****** GX       P$SLOF= ****** GX       RN.NAM= ****** GX           XP.CUG= ****** GX
P$CON = ****** GX       P$MLNK= ****** GX       P$SMA = ****** GX       RN.SCI= ****** GX           XP.DFB= ****** GX
P$COP = ****** GX       P$MNC = ****** GX       P$SPD = ****** GX       RN.SDV= ****** GX           XP.DFW= ****** GX
P$COS = ****** GX       P$MOD = ****** GX       P$STA = ****** GX       RN.SLO= ****** GX           XP.DTA= ****** GX
P$CRID= ****** GX       P$MRC = ****** GX       P$SUBA= ****** GX       RN.SNV= ****** GX           XP.DTE= ****** GX
P$CSR = ****** GX       P$MRES= ****** GX       P$SYS = ****** GX       RN.SPA= ****** GX           XP.KND= ****** GX
P$CT  = ****** GX       P$MRST= ****** GX       P$TLOF= ****** GX       RN.TLO= ****** GX           XP.KNG= ****** GX
P$CTO = ****** GX       P$MUL = ****** GX       P$TRIB= ****** GX       RST     013770R     002 XP.LIN= ****** GX
P$CUG = ****** GX       P$MWI = ****** GX       P$TYP = ****** GX       RTBRK = 000076              XP.MCI= ****** GX
P$DADR= ****** GX       P$MXA = ****** GX       P$URM = ****** GX       RTM     005742R     002 XP.MRE= ****** GX
P$DEA = ****** GX       P$MXCO= ****** GX       P$USE = ****** GX       RTT     007116R     002 XP.MXB= ****** GX
P$DECI= ****** GX       P$MXLI= ****** GX       P$USG = ****** GX       SDBNM   012156R     002 XP.MXC= ****** GX
P$DEF = ****** GX       P$MXR = ****** GX       P$VEC = ****** GX       SF.PH3= 000400              XP.MXR= ****** GX
P$DELI= ****** GX       P$NAM = ****** GX       P$XPRC= ****** GX       STRING  010210R     002 XP.MXW= ****** GX
P$DELO= ****** GX       P$NID = ****** GX       QSTRNG  010230R     002 STRNG   010204R     002 XP.NNA= ****** GX
P$DENO= ****** GX       P$NNA = ****** GX       QUOTE = 000042          STRNG1  010234R     002 XP.NUM= ****** GX
P$DEOB= ****** GX       P$NNO = ****** GX       RDBNM   012062R     002 SW.ENA= ****** GX           XP.RET= ****** GX
P$DEPR= ****** GX       P$NOD = ****** GX       RDBSZ   012120R     002 SW.FUL= ****** GX           XP.RST= ****** GX
P$DFAO= ****** GX       P$NUM = ****** GX       RET     013714R     002 SW.INC= ****** GX           XP.STA= ****** GX
P$DFDA= ****** GX       P$OBJ = ****** GX       RNDCT   003666R     002 SW.LOA= ****** GX           XP.ST1= ****** GX
P$DFWI= ****** GX       P$OID = ****** GX       RNDIA   004256R     002 SW.LON= ****** GX           XS.CLM= ****** GX
P$DIAF= ****** GX       P$ONAM= ****** GX       RNDIA2  004266R     002 SW.OUT= ****** GX           XS.CLV= ****** GX
P$DMOD= ****** GX       P$OUT = ****** GX       RNDLL   003002R     002 SW.PUR= ****** GX           XS.CTM= ****** GX
P$DMPC= ****** GX       P$OVER= ****** GX       RNDMP   004312R     002 SW.RLU= ****** GX           XS.CUG= ****** GX
P$DMPF= ****** GX       P$PAR = ****** GX       RNDMP2  004326R     002 SW.SDI= ****** GX           XS.DST= ****** GX
P$DMPO= ****** GX       P$PEN = ****** GX       RNDPA   003626R     002 SW.SEN= ****** GX           XS.DTE= ****** GX
P$DST = ****** GX       P$PID = ****** GX       RNHAR   003774R     002 SW.SMC= ****** GX           XS.KND= ****** GX
P$DSTO= ****** GX       P$PMOD= ****** GX       RNHAR2  004004R     002 SW.TOP= ****** GX           XS.MXC= ****** GX
P$DTE = ****** GX       P$PNAM= ****** GX       RNHOS   003216R     002 SW.VFY= ****** GX           XS.OBI= ****** GX
P$DTEO= ****** GX       P$POPT= ****** GX       RNLOD   004210R     002 SYSOPT  011530R     002 XS.OBJ= ****** GX
P$DUP = ****** GX       P$PRI = ****** GX       RNLOD2  004220R     002 SYSO    011516R     002 XS.PRI= ****** GX
P$EVE = ****** GX       P$PSTA= ****** GX       RNLOD3  004230R     002 TBIL    014162R     002 XS.SUB= ****** GX
```

```
000202                    25$:    CLOSE$  #$IFDB          ; CLOSE THE INPUT FILE
000212                    30$:    EXST$S  $EXSTS          ; AND GO AWAY...WITH STATUS !
000224                            EXIT$S                  ; JUST IN CASE EXST$S IS NOT SUPPORTED
                                  .DSABL  LSB


          000034'                 .END    $CFE
```

```
CFERQB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:13     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE              REFERENCES

$RLBF      000102 RG        5-102     #6-153
$RQBF      000000 RG       #5-91
$SAVRG   = ******  GX       6-153      6-211
$XXBUF   = ******  GX       5-98
$XXLEN   = ******  GX       5-99      *5-101
.EXINC   = 000020  G       #4-55       5-95         5-100
.RQCB      000206 RG        5-93      #6-211
```

```
CFERQB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:13     PAGE 2
```

```
**FILE**ID**CFESUB
```

```
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE     SSSSSSSS   UU      UU   BBBBBBBB
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE     SSSSSSSS   UU      UU   BBBBBBBB
CC           FF           EE           SS           UU      UU   BB      BB
CC           FF           EE           SS           UU      UU   BB      BB
CC           FF           EE           SS           UU      UU   BB      BB
CC           FF           EE           SS           UU      UU   BB      BB
CC           FFFFFFFF     EEEEEEEE       SSSSSS     UU      UU   BBBBBBBB
CC           FFFFFFFF     EEEEEEEE       SSSSSS     UU      UU   BBBBBBBB
CC           FF           EE                   SS   UU      UU   BB      BB
CC           FF           EE                   SS   UU      UU   BB      BB
CC           FF           EE                   SS   UU      UU   BB      BB       ....
CC           FF           EE                   SS   UU      UU   BB      BB       ....
  CCCCCCCC   FF           EEEEEEEEEE   SSSSSSSS     UUUUUUUUUU   BBBBBBBB         ....
  CCCCCCCC   FF           EEEEEEEEEE   SSSSSSSS     UUUUUUUUUU   BBBBBBBB         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
000530  100434                          BMI     60$             ;NO - TERMINATE
000532  121405                          CMPB    (R4),R5         ;"."?
000534  001407                          BEQ     40$             ;YES
000536  000431                          BR      60$             ;MUST BE A NON-RAD50

000540  005716                  15$:    TST     (SP)            ;EXTENDED MODE?
000542  100427                          BMI     60$             ;NO
000544  012705  000046                  MOV     #35+11,R5       ;ASTERISK USES CODE 35
000550  162705  177767          30$:    SUB     #11-22,R5
000554  162705  177722          40$:    SUB     #22-100,R5
000560  162705  000100          50$:    SUB     #100,R5

                                        .IF DF  R$$EIS

                                        MUL     (R3)+,R5

                                        .IFF

000564  010046                          MOV     R0,-(SP)        ;SAVE NEXT BYTE ADDRESS
000566  012300                          MOV     (R3)+,R0
000570  010146                          MOV     R1,-(SP)        ;SAVE ACCUMULATOR
000572  010501                          MOV     R5,R1
000574  004767  000000G                 CALL    $MUL
000600  010105                          MOV     R1,R5           ;RESULT TO R5
000602  012601                          MOV     (SP)+,R1        ;RESTORE
000604  012600                          MOV     (SP)+,R0

                                        .ENDC

000606  060501                          ADD     R5,R1           ;UPDATE ACCUMULATOR
000610  020327  000422'                 CMP     R3,#CVTBL+6     ;END OF TABLE
000614  103725                          BLO     10$             ;NO - GET NEXT CHAR
000616  005726                          TST     (SP)+           ;CLEAN STACK
000620  000402                          BR      70$

000622  005726                  60$:    TST     (SP)+           ;CLEAN STACK
000624  000261                          SEC
000626  116002  177777          70$:    MOVB    -1(R0),R2       ;RETURN TERMINATING CHAR
000632  000207                          RETURN

                                        .DSABL  LSB
```

```
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE             REFERENCES

T$LLDS     000012          #4-63
T$LLEN     000046          #4-63
T$LOPR     000002          #4-63
T$LTCL     000024          #4-63
T$LTIM     000026          #4-63
T$LTPR     000014          #4-63
T$LTPS     000020          #4-63
T$NAPL     000004          #4-63
T$NFE      000000          #4-63
T$NLEN     000010          #4-63
T$NNUL     000002          #4-63
T$NOPL     000006          #4-63
T$NRNI     000042          #4-63
T$NRPL     000005          #4-63
T$NRUL     000007          #4-63
T$NVR      000001          #4-63
T$RPRI     000040          #4-63
T$SVC      000034          #4-63
T$T5       000030          #4-63
T$T6       000032          #4-63
WC.CNT   = ******   GX      14-506
WC.NAM   = ******   GX      13-460      14-494
WC.TRI   = ******   GX      14-520      15-551
WC.UNT   = ******   GX      14-512
X$LEN      000106           5-100
X$LENX   = 000110   G      #5-100
$CAT5      000442  RG      #12-369
$CFERR     000062  RG      #8-195
$CFER1     000106  RG      #8-202
$C5TA      000244  RG      #11-298
$C5TA1     000234  RG      #11-295
$DIV     = ******   GX      11-321
$DVNAM   = ******   GX      14-486
$ERDPB   = ******   GX      8-203      *8-206      *8-210      8-211      *8-213      8-214      *8-215
$EXSTS   = ******   GX      8-196       8-198      *8-200
$FLAGS   = ******   GX     *13-464     *13-465     *14-498     *14-501     *14-502     *14-505     14-510     14-518
$FNPRO     000634  RG      #13-449
$FNSLT     000730  RG      #14-484
$FNSTA     001172  RG       14-523     #15-544
$MUL     = ******   GX      12-415      16-606
$MVASC     000164  RG      #9-238
$PBLK    = ******   GX      16-613     *16-619
$PRINT     000000  RG      #7-160
$PRNAM   = ******   GX      13-462
$QINS      000200  RG      #10-261
$QINSN     000216  RG      #10-268
$SAVAL   = ******   GX      16-579
$SAVRG   = ******   GX      12-370
$STAN    = ******   GX     *15-555
$TIDPB   = ******   GX     *7-161       7-163
$TMLST   = ******   GX     *10-271      16-580
$UPBLK     001244  RG      #16-578
```

```
    919                                          ; OBJECT OPTIONS
    920                                          ;
    921            000200                                  OO.USR == 200              ; =1 IF OBJECT TASK USER SPECIFIED
    922            000400                                  OO.COP == 400              ; =1 IF OBJECT TASK COPIES SPECIFIED
    923            001000                                  OO.VFY == 1000             ; =1 IF OBJECT TASK VERIFICATION LEVEL SPECIFIED
    924            002000                                  OO.NAM == 2000             ; =1 IF THE OBJECT TASK NAME IS SPECIFIED
    925                                          ;
    926                                          ; LOGGING OPTIONS
    927                                          ;
    928            000001                                  OM.STA == 1                ; =1 IF LOGGING STATE PRESENT
    929            000002                                  OM.MON == 2                ; =1 IF MONITOR SINK TYPE SPECIFIED
    930            000004                                  OM.CON == 4                ; =1 IF CONSOLE SINK TYPE SPECIFIED
    931            000010                                  OM.FIL == 10               ; =1 IF FILE SINK TYPE SPECIFIED
    932            000020                                  OM.EVE == 20               ; =1 IF EVENTS SPECIFIED
    933            000040                                  OM.KEV == 40               ; =1 IF KNOWN EVENTS SPECIFIED
    934                                          ;
    935                                          ; MODULE OPTIONS - X25 PROTOCOL
    936                                          ;
    937                                          ;
    938                                          ;         X25 LEVEL 3 CHARACTERISTICS
    939                                          ;
    940            000001                                  XP.CTM  == 1                        ;CALL TIMER
    941            000002                                  XP.CLT  == 2                        ;CLEAR TIMER
                   000004                                  XP.DFB  == 4                        ;DEFAULT BLOCK
    943            000010                                  XP.DFW  == 10                       ;DEFAULT WINDOW
    944            000020                                  XP.MXB  == 20                       ;MAXIMUM BLOCK
    945            000040                                  XP.MXC  == 40                       ;MAXIMUM CLEARS
    946            000100                                  XP.MRE  == 100                      ;MAXIMUM RESETS
    947            000200                                  XP.MXR  == 200                      ;MAXIMUM RESTARTS
    948            000400                                  XP.MXW  == 400                      ;MAXIMUM WINDOW
    949            001000                                  XP.RET  == 1000                     ;RESET TIMER
    950            002000                                  XP.RST  == 2000                     ;RESTART TIMER
    951            004000                                  XP.NNA  == 4000                     ;NETWORK NAME (PSN$DF)
    952
    953            000177                                  XP.LV3  == XP.CTM!XP.CLT!XP.DFB!XP.DFW!XP.MXB!XP.MXC!XP.MRE
    954            003777                                  XP.LV3  == XP.LV3!XP.MXR!XP.MXW!XP.RET!XP.RST
    955                                          ;
    956                                          ;         DTE PARAMETERS
    957                                          ;
    958            010000                                  XP.DTE  == 10000                    ;DTE
    959            020000                                  XP.KND  == 20000                    ;KNOWN DTES
    960                                          ; SUB-PARAMETERS SPECIFIC TO DTE ($SWICH)
    961            000001                                     XP.CHN  == 1                     ;CHANNEL
    962            000002                                     XP.COU  == 2                     ;COUNTER TIMER
    963            000004                                     XP.LIN  == 4                     ;LINE
    964            000010                                     XP.MCI  == 10                    ;MAXIMUM CIRCUITS
    965            000020                                     XP.STA  == 20                    ;STATE
    966            000040                                     XP.ST1  == 40                    ;STATE = ON
    967                                          ;
    968                                          ;         GROUP PARAMETERS
    969                                          ;
    970            040000                                  XP.CUG  == 40000                    ;USER GROUP
    971            100000                                  XP.KNG  == 100000                   ;KNOWN GROUPS
    972                                          ; SUB-PARAMETERS SPECIFIC TO CUG ($SWICH)
    973            000001                                     XP.DTA  == 1                     ;DTE ADDRESS
    974            000002                                     XP.NUM  == 2                     ;NUMBER
    975            000004                                     XP.BIL  == 4                     ;TYPE BILATERAL
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 3
SYMBOL CROSS REFERENCE                                   CREF   04.00
SYMBOL   VALUE              REFERENCES

C$UNAM     000004  G        #9-547
C.DSDS   = 000006            6-95
C.LNK      000000  G        #12-1041
C.PRM      000004  G        #12-1043
C.SIZE   = 000054            6-95
C.STS      000002  G        #12-1042
DA$CUG   = 004000  G        #12-1176
DEFFLG   = 000233           #6-95         6-95
DFLTNB     000426 R          6-95        #6-95
DFTLEN   = 000007           #6-95
D$ACUG     000022  G        #9-632
D$ADLN     000026  G        #9-633
D$ALEN   = 000030  G        #9-634
D$ANUM     000004  G        #9-629
D$ASHI     000020  G        #9-631
D$ASLO     000016  G        #9-630
D$CLEN   = 000062  G        #9-649
D$CMLN     000060  G        #9-647
D$CMSK     000004  G        #9-645
D$CVAL     000032  G        #9-646
D$CVLN     000061  G        #9-648
D$DFLG     000006  G        #9-374
D$DLEN   = 000014  G        #9-378
D$DNAM     000004  G        #9-373
D$DNCT     000013  G        #9-377
D$DNLT     000012  G        #9-376
D$DPRI     000010  G        #9-375
D$SLEN   = 000020  G        #9-619
D$SNAM     000006  G        #9-615
D$SOBJ     000013  G        #9-617
D$SPRI     000012  G        #9-616
D$STSK     000014  G        #9-618
D$STYP     000004  G        #9-614
D$TADR     000004  G        #9-516
D$TCT      000032  G        #9-521
D$THSH     000030  G        #9-520
D$TLEN   = 000040  G        #9-523
D$TLIN     000016  G        #9-517
D$TNET     000034  G        #9-522
D$TSTA     000026  G        #9-519
EF       = 000001           #6-95         6-95          6-95          6-95
ERRALL     002336 RG        #13-1221
ERRCIS     002507 RG        #13-1224
ERRCON     002101 RG        #13-1215
ERRFIL     002245 RG        #13-1219
ERRGCL     002163 RG        #13-1217
ERRILC     002133 RG        #13-1216
ERRILN     002216 RG        #13-1218
ERRLIS     002445 RG        #13-1223
ERRNOD     002554 RG        #13-1225
ERROUT     002303 RG        #13-1220
ERRPNA     002616 RG        #13-1226
```

```
                                                                              J  3
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 16
SYMBOL CROSS REFERENCE                                   CREF   04.00
SYMBOL  VALUE               REFERENCES

                        #6-113      6-113     #6-113      6-113     #6-113      6-113     #6-115      6-115      6-115
                        #6-115      6-115      6-115     #6-115      6-115
...PC3  = 000142 R      #6-95       6-95      #6-115      6-115
```

K 3

```
                                ;****
                                ; LIBRARY MACRO CALLS
                                ;****

                                        .MCALL  QIOW$,DIR$,GTIM$
                                ;****
                                ; LOCAL DATA
                                ;****

                                ;
                                ; DISPATCH TABLE
                                ;
 000000                         DSPTBL:
 000000  000000G                        .WORD   $SHREM          ; "SHOW NODE" REQUEST
 000002  000000G                        .WORD   $SHPRO          ; "SHOW PROCESS" REQUEST
 000004  000000G                        .WORD   $SHEXE          ; "SHOW EXECUTOR" REQUEST
 000006  000000G                        .WORD   $SHSYS          ; "SHOW SYSTEM" REQUEST
 000010  000000G                        .WORD   $SHOBJ          ; "SHOW OBJECT" REQUEST
 000012  000000G                        .WORD   $SHLIN          ; "SHOW LINE" REQUEST
 000014  000000G                        .WORD   $SHLOG          ; "SHOW LOGGING" REQUEST
 000016  000000G                        .WORD   $SHCIR          ; "SHOW CIRCUIT" REQUEST
 000020  000000G                        .WORD   $SHCIR          ; "SHOW KNOWN CIRCUITS" REQUEST
 000022  000000G                        .WORD   $SHXPR          ; "SHOW MODULE X25-PROTOCOL" REQUEST
 000024  000000G                        .WORD   $SHXSV          ; "SHOW MODULE X25-SERVER" REQUEST
 000026  000000G                        .WORD   $SHXAC          ; "SHOW MODULE X25-ACCESS" REQUEST
 000030  000000G                        .WORD   $SHX29          ; "SHOW MODULE X29-SERVER" REQUEST
 000032  000000G                        .WORD   $SHSNA          ; "SHOW MODULE SNA-PROTOCOL" REQUEST

                                ;
                                ; QIO DPB FOR ATTACHING AND DETACHING FROM TERMINAL
                                ;
 000034                         TIDPB:  QIOW$   IO.ATT,2,2
 000064                         LFDPB:  QIOW$   IO.WVB,2,2,,,,<BLANK,1,40>
 000114                         GTIM:   GTIM$   $GTBUF
 000120                         $HDRBK::.BLKW   4.              ; HEADER PARAMETERS
 000130                         $GTBUF::.BLKW   8.              ; GET TIME BUFFER
 000150      040                BLANK:  .ASCII  / /
                                        .EVEN
```

```
CFEOPN     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE             REFERENCES

BUFSZ   = 000122           #4-76     4-77      4-78
CSBLK     000300 R         #4-79     5-98      5-107     7-160     7-171     7-172    *7-179     7-181    *7-183
CS.OUT  = 000002            5-103    7-165
C.CMLD  = 000002           *5-99    *5-100    *7-161    *7-162
C.DSDS  = 000006            5-107    7-181
C.FILD  = 000016            7-171    7-172    *7-179    *7-183
C.SIZE  = 000054            4-79     5-101     5-103     7-163     7-165
FO.RD   = ******   GX       5-108
FO.WRT  = ******   GX       7-182
F.DSPT  = 000044           *5-107   *7-181
F.ERR   = 000052           *5-105   *7-166
F.FACC  = 000043           *5-108   *7-182
F.FNB   = 000102            4-73
F.LUN   = 000042            5-113
F.NRBD  = 000024            6-139
F.URBD  = 000020           *6-137
IE.BNM  = ******   GX       5-105    7-166
IOSB      000030 R          4-73    #4-74      5-116
IO.RAT  = ******   GX       4-73
N.FID   = 000000            4-73
Q.IOFN  = 000002            4-73
Q.IOLU  = 000004           *5-113
RATDPB    000000 R         #4-73    *5-113     5-114
$BUFI     000156 RG        #4-78     6-137    *6-140
$BUFR     000034 RG        #4-77
$FIATR  = ******   GX       4-73    *5-118
$IFDB   = ******   GX       4-73     5-104     6-136
$IOPEN    000354 RG        #5-97
$OFDB   = ******   GX      *7-166    7-180     8-201
$OPEN     000530 RG        #7-158
$RDATA    000476 RG        #6-135
$WDATA    000662 RG        #8-200
$$$ARG  = 000002           #4-73     4-73     #4-73      4-73     #4-73      4-73      4-73
$$$GLB  = ******            4-73
.CSI1   = ******   G        5-101    5-101     7-163     7-163
.CSI2   = ******   G        5-103    5-103     7-165     7-165
.GET    = ******   G        6-137    6-137
.OPEN   = ******   G        5-108    5-108     7-182     7-182
.PUT    = ******   G        8-202    8-202
```

```
                                                .SBTTL  PURCIR  - PURGE CIRCUIT
                                        ;+
                                        ; ** PURLIN - PURGE CIRCUIT
                                        ;
                                        ; INPUTS        $CIRID = CIRCUIT ID IN PARSED DATA AREA
                                        ;               $FLAGS = CIRCUIT ATTRIBUTES (SVC OR PVC)
                                        ;               $COMP  = COMPONENT (CIRCUIT OR KNOWN CIRCUITS)
                                        ;
                                        ; OUTPUTS       THE CIRCUIT IS PURGED.
                                        ;-

 000756                                 PURCIR:
                                        ;
                                        ; CHECK CIRCUIT TYPE
                                        ;
 000756  010605                                 MOV     SP,R5                   ; ASSUME WILDCARD
 000760  022767  000000G 000000G                CMP     #CO.KNC,$COMP           ; KNOWN CIRCUITS ?
 000766  001411                                 BEQ     10$                     ; BR IF YES
 000770  005005                                 CLR     R5                      ; ELSE NOTHING FOUND YET
 000772  032767  000000G 000000G                BIT     #FL.X25,$FLAGS          ; IS THIS AN X25 CIRCUIT
 001000  001507                                 BEQ     101$                    ; BR IF WRONG TYPE OF CIRCUIT
 001002  032767  000000G 000000G                BIT     #FL.DLM,$FLAGS          ; CHECK FOR ILLEGAL CIRCUIT TYPES
 001010  001103                                 BNE     101$                    ; BR IF ERRORS
                                        ;
                                        ; LOOK FOR CIRCUIT DESCRIPTOR
                                        ;
 001012  012702  000000G                10$:    MOV     #$TMLST,R2              ; POINT TO LISTHEAD
 001016  010203                         20$:    MOV     R2,R3                   ; SAVE PREVIOUS ADDRESS
 001020  011202                                 MOV     (R2),R2                 ; NEXT TEMPLATE
 001022  001473                                 BEQ     70$                     ; IF EQ, END OF LIST
 001024  012701  000000G                        MOV     #P$VLEN,R1              ; LENGTH FOR PVC
 001030  012704  000000G                        MOV     #P$VNAM,R4              ; OFFSET OF NAME
 001034  122762  000000G 000000G                CMPB    #CS.PVC,C.STS(R2);      PVC ?
 001042  001365                                 BNE     20$                     ; IF NE, KEEP ON LOOKING
 001044  022767  000000G 000000G        30$:    CMP     #CO.KNC,$COMP           ; WILDCARD ?
 001052  001407                                 BEQ     40$                     ; BR IF YES
 001054  060204                                 ADD     R2,R4                   ; POINT TO NAME
 001056  022467  000000G                        CMP     (R4)+,$CIRID            ; CHECK CIRCUIT ID
 001062  001355                                 BNE     20$                     ; IF NE WRONG CIRCUIT
 001064  021467  000002G                        CMP     (R4),$CIRID+2           ;   ...
 001070  001352                                 BNE     20$                     ; BR IF WRONG ONE
 001072  005205                         40$:    INC     R5                      ; REMEMBER FOUND
 001074  032767  000000G 000000G                BIT     #OC.CTM,$OPRND          ; PURGE COUNTER TIMER?
 001102  001030                                 BNE     50$                     ; IF NE, YES
                                        ;
                                        ; RETURN CIRCUIT BLOCK TO FREE LIST
                                        ;
 001104  032767  000000G 000000G                BIT     #FL.DLM,$FLAGS          ; DLM CIRCUIT ?
 001112  001341                                 BNE     20$                     ; IF NE, YES - THIS ONE DOESN'T GET PURGED
 001114  026227  000000G 015355                 CMP     P$VOWN(R2),#^RDLM       ; DLM PVC ?
 001122  001735                                 BEQ     20$                     ; IF EQ, NOT THIS ONE
 001124  011213                                 MOV     (R2),(R3)               ; CLOSE UP LIST
 001126  001002                                 BNE     45$                     ; IF NE, NOT LAST
 001130  010367  000002G                        MOV     R3,$TMLST+2             ; FIX UP LISTHEAD
 001134  012700  000000G                45$:    MOV     #$XXAVL,R0              ; POINT TO FREE LISTHEAD
 001140  004767  000000G                        CALL    $RLBF                   ; RELEASE CORE BLOCK
 001144  010302                                 MOV     R3,R2                   ; PICK UP WHERE WE LEFT OFF
```

```
000000                        KEYWD$  ''TYPE'',          K$TYP
000000                        KEYWD$  ''UBE'',           K$UBE
000000                        KEYWD$  ''UBF'',           K$UBF
000000                        KEYWD$  ''UBH'',           K$UBH
000000                        KEYWD$  ''UBJ'',           K$UBJ
000000                        KEYWD$  ''UBK'',           K$UBK
000000                        KEYWD$  ''UBL'',           K$UBL
000000                        KEYWD$  ''UBM'',           K$UBM
000000                        KEYWD$  ''UBN'',           K$UBN
000000                        KEYWD$  ''UBP'',           K$UBP
000000                        KEYWD$  ''UBR'',           K$UBR
000000                        KEYWD$  ''UBS'',           K$UBS
000000                        KEYWD$  ''UBT'',           K$UBT
000000                        KEYWD$  ''UNIT'',          K$UNT
000000                        KEYWD$  ''URM'',           K$URM
000000                        KEYWD$  ''USAGE'',         K$USG
000000                        KEYWD$  ''USER'',          K$USE
000000                        KEYWD$  ''VALUE'',         K$VAL
000000                        KEYWD$  ''VECTOR'',        K$VEC
000000                        KEYWD$  ''VERIFICATION'',  K$VER
000000                        KEYWD$  ''VERSION'',       K$VRSN
000000                        KEYWD$  ''WINDOW'',        K$WIN
000000                        KEYWD$  ''X25'',           K$X25
000000                        KEYWD$  ''X29'',           K$X29
000000                        KEYWD$  ''III'',           K$3
000000                        KEYWD$  ''IV'',            K$4

                      ;
                      ; SERVICE DEVICE TYPES
                      ;
000000                        KEYWD$  ''DP'',    K$XDP
000000                        KEYWD$  ''UNA'',   K$XUNA
000000                        KEYWD$  ''DU'',    K$XDU
000000                        KEYWD$  ''CNA'',   K$XCNA
000000                        KEYWD$  ''DL'',    K$XDL
000000                        KEYWD$  ''QNA'',   K$XQNA
000000                        KEYWD$  ''DQ'',    K$XDQ
000000                        KEYWD$  ''CI'',    K$XCI
000000                        KEYWD$  ''DA'',    K$XDA
000000                        KEYWD$  ''PCL'',   K$XPCL
000000                        KEYWD$  ''DUP'',   K$XDUP
000000                        KEYWD$  ''DMC'',   K$XDMC
000000                        KEYWD$  ''DN'',    K$XDN
000000                        KEYWD$  ''DLV'',   K$XDLV
000000                        KEYWD$  ''DMP'',   K$XDMP
000000                        KEYWD$  ''DTE'',   K$XDTE
000000                        KEYWD$  ''DV'',    K$XDV
000000                        KEYWD$  ''DZ'',    K$XDZ
000000                        KEYWD$  ''KDP'',   K$XKDP
000000                        KEYWD$  ''KDZ'',   K$XKDZ
000000                        KEYWD$  ''KL'',    K$XKL
000000                        KEYWD$  ''DMV'',   K$XDMV
000000                        KEYWD$  ''DPV'',   K$XDPV
000000                        KEYWD$  ''DMF'',   K$XDMF
000000                        KEYWD$  ''DMR'',   K$XDMR
000000                        KEYWD$  ''KMY'',   K$XKMY

                              .SBTTL  COMMANDS
```

```
000000                          TRAN$   $LAMDA, $FAIL,  $CLERR, ERRIPV, $ERROR
                        ;
                        ; Dump address
                        ;
000000                          STATE$  RNDPA
000000                           TRAN$  &K$DMP
000000                          STATE$  ,&P$DMPO
000000                           TRAN$  &K$ADD
000000                          STATE$  ,&P$DADR
000000                           TRAN$  $NUMBR, $EXIT,  .RNDPA
000000                           TRAN$  $LAMDA, $FAIL,  $CLERR, ERRILN, $ERROR
                        ;
                        ; Dump Count
                        ;
000000                          STATE$  RNDCT
000000                           TRAN$  &K$DMP
000000                          STATE$  ,&P$DMPO
000000                           TRAN$  &K$COUN
000000                          STATE$  ,&P$DMPC
000000                           TRAN$  $NUMBR, $EXIT,  .RNDCT
000000                           TRAN$  $LAMDA, $FAIL,  $CLERR, ERRILN, $ERROR
                        ;
                        ; [Service] circuit
                        ;
000000                          STATE$  RNSCI
000000                           TRAN$  &K$CIR, RNSCI2
000000                           TRAN$  &K$SER
000000                          STATE$  ,&P$SERO
000000                           TRAN$  &K$CIR
000000                          STATE$  RNSCI2, &P$SERC
000000                           TRAN$  $LAMDA,,,1,$CIRCT
000000                          STATE$
000000                           TRAN$  !LINCIR,        $EXIT,  .RNSCI
000000                           TRAN$  $LAMDA, $FAIL
                        ;
                        ; Hardware [address]
                        ;
000000                          STATE$  RNHAR
000000                           TRAN$  &K$HRD
000000                          STATE$
000000                           TRAN$  &K$ADD,RNHAR2
000000                           TRAN$  $LAMDA
000000                          STATE$  RNHAR2,&P$HADR
000000                           TRAN$  !HRDSTR,$EXIT,  .RNHAR
000000                           TRAN$  $LAMDA, $FAIL,  $CLERR, ERRILN, $ERROR

000000                          STATE$  HRDSTR
000000                           TRAN$  $LAMDA,,        $BL.MAN
000000                          STATE$  HRDST2
000000                           TRAN$  $ALPHA, HRDST2
000000                           TRAN$  $DIGIT, HRDST2
000000                           TRAN$  <'->,   HRDST2
000000                           TRAN$  $LAMDA, $EXIT,  $BL.AUT

                        ;
                        ; [Service] password
                        ;
```

```
000000                                   TRAN$   <'*>,,,WC.CNT,$WILD

000000                                  STATE$   LIN1
000000                                   TRAN$   <'->,LIN2
000000                                   TRAN$   $LAMDA,LIN3,.XUNT

000000                                  STATE$   LIN2
000000                                   TRAN$   $DNUMB,LIN3,.LIUNT
000000                                   TRAN$   <'*>,,,WC.UNT,$WILD

000000                                  STATE$   LIN3
000000                                   TRAN$   '.,LIN4
000000                                   TRAN$   $LAMDA,$EXIT,.LIEND

000000                                  STATE$   LIN4
000000                                   TRAN$   !LIN5,,,FL.TRI,$FLAGS
000000                                  STATE$
000000                                   TRAN$   $LAMDA,$EXIT,.LIEND

000000                                  STATE$   LIN5
000000                                   TRAN$   $DNUMB,$EXIT,.LISTA
000000                                   TRAN$   <'*>,$EXIT,,WC.TRI,$WILD

                          ;
                          ; LOOK FOR ''KNOWN LINES''
                          ;
000000                                  STATE$   KNLIN
000000                                   TRAN$   &K$KNO

000000                                  STATE$    ,&P$PEN
000000                                   TRAN$   &K$LINS,$EXIT,,WC.LIN,$WILD
                          ;
                          ; LOOK FOR ''KNOWN CIRCUITS''
                          ;
000000                                  STATE$   KNCIR
000000                                   TRAN$   &K$KNO

000000                                  STATE$    ,&P$PEN
000000                                   TRAN$   &K$CIRS,$EXIT,,WC.CIR,$WILD
```

```
                                ;
000000                                   STATE$  DSTLIS
000000                                    TRAN$  !DST,     DSTLS1,,XS.DST, $OPRND
000000                                    TRAN$  !KNDS,,,          XS.KND, $OPRND
000000                                   STATE$  DSTLS1, &P$DSTO
000000                                    TRAN$  $LAMDA
000000                                   STATE$  DSTLS2
000000                                    TRAN$  !DSTLSO,DSTLS2
000000                                    TRAN$  $LAMDA, $EXIT
000000                                   STATE$  DSTLSO
000000                                    TRAN$  !CLM,    $EXIT,, XS.CLM, $OPRND
000000                                    TRAN$  !CLV,    $EXIT,, XS.CLV, $OPRND
000000                                    TRAN$  !CUG,    $EXIT,, XS.CUG, $OPRND
000000                                    TRAN$  !DNUM,   $EXIT,, XS.DTE, $OPRND
000000                                    TRAN$  !OBJN,   $EXIT,, XS.OBJ, $OPRND
000000                                    TRAN$  !OBJI,   $EXIT,, XS.OBI, $OPRND
000000                                    TRAN$  !DPRI,   $EXIT,, XS.PRI, $OPRND
000000                                    TRAN$  !DSUB,   $EXIT,, XS.SUB, $OPRND
                                ;
                                ; DEFINE MODULE X25-PROTOCOL CHARACTERISTICS OPTIONS
                                ;
000000                                   STATE$  DXPRCH, &P$XPRC
000000                                    TRAN$  !DXPRC2
000000                                   STATE$  DXPRC1
000000                                    TRAN$  $EOS,    $EXIT
000000                                    TRAN$  !DXPRC2,DXPRC1

000000                                   STATE$  DXPRC2
000000                                    TRAN$  !CTM,    $EXIT,, XP.CTM, $OPRND
000000                                    TRAN$  !CLT,    $EXIT,, XP.CLT, $OPRND
000000                                    TRAN$  !DFB,    $EXIT,, XP.DFB, $OPRND
000000                                    TRAN$  !DFW,    $EXIT,, XP.DFW, $OPRND
000000                                    TRAN$  !MMDA,   $EXIT,, XP.MXB, $OPRND
000000                                    TRAN$  !MWI,    $EXIT,, XP.MXW, $OPRND
000000                                    TRAN$  !MXC,    $EXIT,, XP.MXC, $OPRND
000000                                    TRAN$  !MRE,    $EXIT,, XP.MRE, $OPRND
000000                                    TRAN$  !RET,    $EXIT,, XP.RET, $OPRND
000000                                    TRAN$  !MXR,    $EXIT,, XP.MXR, $OPRND
000000                                    TRAN$  !RST,    $EXIT,, XP.RST, $OPRND
000000                                    TRAN$  !NNAM,   $EXIT,, XP.NNA, $OPRND
                                ;
                                ; X25-PROTOCOL DTE PARAMETERS
                                ;
000000                                   STATE$  DXPDDT, &P$XPRC
000000                                    TRAN$  !KND,    DXPDDO,,XP.KND, $OPRND
000000                                    TRAN$  !DTE,          ,,XP.DTE, $OPRND
000000                                   STATE$  DXPDDO, &P$DTEO
000000                                    TRAN$  !DXPDD2
000000                                   STATE$  DXPDD1
000000                                    TRAN$  $EOS,    $EXIT
000000                                    TRAN$  !DXPDD2,DXPDD1

000000                                   STATE$  DXPDD2
000000                                    TRAN$  !CT,     $EXIT,, XP.COU, $SWICH
000000                                    TRAN$  !LINID,  $EXIT,, XP.LIN, $SWICH
000000                                    TRAN$  !MCIRD,  $EXIT,, XP.MCI, $SWICH
000000                                    TRAN$  !DSTA,   $EXIT,, XP.STA, $SWICH
```

```
                                                 J 11
CFEPRS - STATE TABLES TO PARSE  MACRO V05.03b Tuesday 02-Jul-85 14:08  Page 15-4
Symbol table

$ALPHA= 000022            $SNDEF= ****** GX         .EXBEA= ****** GX        .LISTA= ****** GX        .OBNUM= ****** GX
$ANY  = 000020            $SNLIS= ****** GX         .EXBRT= ****** GX        .LIUNT= ****** GX        .PRMWI= ****** GX
$BAUD = ****** GX         $SNPUR= ****** GX         .EXCOS= ****** GX        .LIVEC= ****** GX        .PRNAM= ****** GX
$BLANK= 000006            $STRNG= 000004            .EXHOP= ****** GX        .LIX25= ****** GX        .PRNCT= ****** GX
$BL.AU= ****** GX         $SUBXP= 000010            .EXHOS= ****** GX        .MOCAT= ****** GX        .PRNLT= ****** GX
$BL.A3= ****** GX         $SWICH= ****** GX         .EXHO1= ****** GX        .MOCLM= ****** GX        .RNADD= ****** GX
$BL.MA= ****** GX         $URM  = ****** GX         .EXID = ****** GX        .MOCLT= ****** GX        .RNAD2= ****** GX
$BL.MN= ****** GX         $VLVL = ****** GX         .EXID1= ****** GX        .MOCLV= ****** GX        .RNAD3= ****** GX
$CIRCT= ****** GX         $WILD = ****** GX         .EXINA= ****** GX        .MOCNN= ****** GX        .RNCHK= ****** GX
$CLERR= ****** GX         $$$FLG= 177777            .EXINT= ****** GX        .MOCRN= ****** GX        .RNCLF= ****** GX
$CMAND= ****** GX         $$$KEY= 000274            .EXLNK= ****** GX        .MOCUG= ****** GX        .RNDCT= ****** GX
$COMP = ****** GX         $$$LEN= 000006            .EXNAM= ****** GX        .MODFD= ****** GX        .RNDPA= ****** GX
$DIGIT= 000024            $$$STA= 000000            .EXNC = ****** GX        .MODFW= ****** GX        .RNFSP= ****** GX
$DNUMB= 000014            $$$TMP= 002303R       004 .EXNN = ****** GX        .MODST= ****** GX        .RNHAR= ****** GX
$EOS  = 000012            .BL.AU= ****** GX         .EXOUT= ****** GX        .MODS1= ****** GX        .RNHNA= ****** GX
$ERROR= ****** GX         .CIACT= ****** GX         .EXRET= ****** GX        .MODS2= ****** GX        .RNHOS= ****** GX
$EXIT = 000000            .CICHN= ****** GX         .EXRTM= ****** GX        .MOINC= ****** GX        .RNHO1= ****** GX
$FAIL = 177777            .CICOS= ****** GX         .EXSB2= ****** GX        .MOMCI= ****** GX        .RNHO2= ****** GX
$FLAGS= ****** GX         .CICT = ****** GX         .EXSEG= ****** GX        .MOMCL= ****** GX        .RNINI= ****** GX
$GOFF = 000001            .CIDTE= ****** GX         .EXSUB= ****** GX        .MOMC1= ****** GX        .RNIN2= ****** GX
$GPRM = 000001            .CIHTM= ****** GX         .HLPAC= ****** GX        .MOMDA= ****** GX        .RNNAM= ****** GX
$INFO = ****** GX         .CIMBL= ****** GX         .LICNT= ****** GX        .MOMRE= ****** GX        .RNSCI= ****** GX
$LAMDA= 000000            .CIMRC= ****** GX         .LICSR= ****** GX        .MOMXR= ****** GX        .RNSDV= ****** GX
$NUMBR= 000002            .CIMXR= ****** GX         .LICTM= ****** GX        .MONNA= ****** GX        .RNSPA= ****** GX
$OPRND= ****** GX         .CIRID= ****** GX         .LIDEA= ****** GX        .MONUM= ****** GX        .SYBYT= ****** GX
$OPRN2= ****** GX         .CIRI1= ****** GX         .LIEND= ****** GX        .MOOBJ= ****** GX        .SYCBN= ****** GX
$PRGCL= ****** GX         .CIRPR= ****** GX         .LIHBT= ****** GX        .MOOBN= ****** GX        .SYPAR= ****** GX
$RAD50= 000016            .CIRTM= ****** GX         .LIMWI= ****** GX        .MOPRI= ****** GX        .SYPOL= ****** GX
$RNOIA= ****** GX         .CITRI= ****** GX         .LINAM= ****** GX        .MORET= ****** GX        .SYRBN= ****** GX
$RNDMP= ****** GX         .CKDLM= ****** GX         .LIPRI= ****** GX        .MORST= ****** GX        .SYRBS= ****** GX
$RNLOD= ****** GX         .EACOS= ****** GX         .LIRTC= ****** GX        .NONAM= ****** GX        .SYSBN= ****** GX
$RNSLO= ****** GX         .EAHOP= ****** GX         .LIRTT= ****** GX        .OBCOP= ****** GX        .SYTHR= ****** GX
$RNTLO= ****** GX         .EAMAX= ****** GX         .LISEC= ****** GX        .OBNAM= ****** GX        .XUNT = ****** GX
$RONLY= 000001            .EXADD= ****** GX         .LISPD= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
$STATE  015062    002   (RO,D,LCL,REL,CON)
$KTAB   000572    003   (RO,D,LCL,REL,CON)
$KSTR   002307    004   (RO,D,LCL,REL,CON)
$PRMPT  000000    005   (RO,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10744 Words  ( 42 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:29:33.36
SY:CFEPRS.V1,[132,134]CFEPRS/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEPRS
```

K 11

```
                                                                                     J 12
CFEROT - ROOT SEGMENT FOR CFE   MACRO V05.03b Saturday 29-Jun-85 05:13  Page 5-2
Symbol table

ERRCON= ****** GX         F.DSPT= 000044            F.RACC= 000016            N.FTYP= 000014            S.FTYP= 000002
EX$ERR= 000002            F.DVNM= 000134            F.RATT= 000001            N.FVER= 000016            S.NFEN= 000020
EX$SEV= 000004            F.EFBK= 000010            F.RCNM= 000034            N.NEXT= 000022            WATDPB  000000R
EX$SUC= 000001            F.EFN = 000050            F.RCTL= 000017            N.STAT= 000020            $CFCOM= ****** GX
EX$WAR= 000000            F.EOBB= 000032            F.RSIZ= 000002            N.UNIT= 000034            $CFDSP= ****** GX
FX.DIR= 000004            F.ERR = 000052            F.RTYP= 000000            PAR$$$= 000027            $CFE    000034RG
FX.DLN= 000003            F.EXT = 000072            F.SEQN= 000100            Q.IOAE= 000012            $CFERR= ****** GX
FX.DMX= 000002            F.FACC= 000043            F.SPDV= 000072            Q.IOEF= 000006            $CFINI= ****** GX
FX.LEN= 000000            F.FFBY= 000014            F.SPUN= 000074            Q.IOFN= 000002            $CFUPD= ****** GX
F.ACTL= 000076            F.FLG = 000074            F.STBK= 000036            Q.IOLU= 000004            $EXSTS= ****** GX
F.ALOC= 000040            F.FNAM= 000110            F.UNIT= 000136            Q.IOPL= 000014            $FIATR= ****** GX
F.BBFS= 000062            F.FNB = 000102            F.URBD= 000020            Q.IOPR= 000007            $FIN    000056RG
F.BDB = 000070            F.FTYP= 000116            F.VBN = 000064            Q.IOSB= 000010            $IFDB = ****** GX
F.BGBC= 000057            F.FVER= 000120            F.VBSZ= 000060            SW.PUR= ****** GX         $OFDB = ****** GX
F.BKDN= 000026            F.HIBK= 000004            IOSB    000030R           S.ANM1= 000014            $SWICH= ****** GX
F.BKDS= 000020            F.LUN = 000042            IO.WAT= ****** GX         S.ANM2= 000005            $$$ARG= 000002
F.BKEF= 000050            F.MBCT= 000054            N.ANM1= 000002            S.FATT= 000016            $$$OST= 000005
F.BKP1= 000051            F.MBC1= 000055            N.ANM2= 000024            S.FDB = 000140            .CLOSE= ****** G
F.BKST= 000024            F.MBFG= 000056            N.DID = 000024            S.FNAM= 000006            .CONER  000162RG
F.BKVB= 000064            F.NRBD= 000024            N.DVNM= 000032            S.FNB = 000036            .DELET= ****** G
F.CHR = 000075            F.NREC= 000030            N.FID = 000000            S.FNBW= 000017            .EXIT   000172RG
F.CNTG= 000034            F.OVBS= 000030            N.FNAM= 000006            S.FNTY= 000004            ...TPC= 000006
F.DFNB= 000046

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000232    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9848 Words  ( 39 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:12.57
SY:CFEROT.V2,[132,134]CFEROT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEROT
```

K 12

```
CFERQB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:13     PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME        REFERENCES

DIR$              #5-95        5-95
EXTK$S            #4-50        5-95
MOV$              #5-95        5-95
RESRG             #4-50        5-103
SAVRG             #4-50        5-92
```

Table of contents

```
     7-   147     $PRINT - PRINT ASCIZ STRING
     8-   178     $CFERR - SET UP EXIT STATUS AND PRINT ASCIZ ERROR MESSAGE
     8-   179     $CFER1 - PRINT ASCIZ STRING ON ERROR LUN
     9-   219     $MVASC - MOVE ASCII STRING
    10-   246     $QINS - INSERT ENTRY AT END OF QUEUE
    10-   247     $QINSN - INSERT ENTRY INTO MIDDLE OF QUEUE
    11-   275     $C5TA - CONVERT EXTENDED RAD50 TO ASCII ("*")
    11-   276     $C5TA1- CONVERT EXTENDED RAD50 TO ASCII ("-")
    12-   338     $CAT5 - ASCII TO RAD50 CONVERSION
    12-   339     .CAT5 - ASCII TO EXTENDED RAD50 CONVERSION ("*")
    12-   340     .CAT5E- ASCII TO EXTENDED RAD50 CONVERSION ("-")
    13-   435     $FNPRO - FIND A PROCESS IN THE TEMPLATES
    14-   468     $FNSLT - FIND AN SLT TEMPLATE
    15-   530     $FNSTA - FIND AN STATION TEMPLATE
    16-   560     $UPBLK - UPDATE POOL BLOCK ALLOCATION
```

```
                                                  .SBTTL  $FNPRO - FIND A PROCESS IN THE TEMPLATES
                                          ;+
                                          ; *** - $FNPRO - FIND A PROCESS IN THE TEMPLATES
                                          ;
                                          ; INPUT:
                                          ;       $PRNAM = THE PROCESS NAME IN RAD50
                                          ;       R0 = POINTER TO CURRENT TEMPLATE
                                          ;
                                          ; OUTPUT:
                                          ;       C-BIT = SUCCESS/FAILURE
                                          ;       R0 = ADDRESS OF TEMPLATE
                                          ;
                                          ;-

 000634                                   $FNPRO::
 000634  011000                           10$:    MOV     (R0),R0                 ; GET NEXT TEMPLATE
 000636  000261                                   SEC                             ; ASSUME FAILURE
 000640  001431                                   BEQ     20$                     ; IF EQ, END OF LIST

 000642  122760  000000G 000000G                  CMPB    #CS.LLC,C.STS(R0)       ; IS THIS A PROCESS TEMPLATE?
 000650  001410                                   BEQ     12$                     ; BR IF YES
 000652  122760  000000G 000000G                  CMPB    #CS.DLC,C.STS(R0)       ; ...
 000660  001404                                   BEQ     12$                     ; BR IF YES
 000662  122760  000000G 000000G                  CMPB    #CS.DDM,C.STS(R0)       ; ...
 000670  001361                                   BNE     10$                     ; BR IF NO .. KEEP LOOKING
 000672  032767  000000G 000000G  12$:            BIT     #WC.NAM,$WILD           ; ALL PROCESSES ?
 000700  001004                                   BNE     15$                     ; IF NE, YES
 000702  026760  000000G 000000G                  CMP     $PRNAM,C.PRM(R0)        ; IS THIS THE RIGHT PROCESS ?
 000710  001351                                   BNE     10$                     ; IF NE, NO .. KEEP LOOKING
 000712  105067  000000G          15$:            CLRB    $FLAGS                  ; CLEAR PROCESS FLAGS (CLEARS C-BIT)
 000716  156067  000002G 000000G                  BISB    C.PRM+2(R0),$FLAGS      ; GET FLAGS
 000724  000207                   20$:            RETURN
```

```
CFESUB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 5
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE               REFERENCES

$WILD    = ******  GX         13-460     14-494     14-506     14-512     14-520     15-551
.CAT5      000432 RG         #12-366
.CAT5E     000422 RG         #12-363
.FNSLT     000726 RG         #14-483
```

```
                                ;
                                ; MODULE OPTIONS - X25 SERVER
                                ;
        000020                          XS.STA  == XP.STA               ;STATE  ($SWICH)
        000040                          XS.ST1  == XP.ST1               ;STATE = ($SWICH)

        000001                          XS.DST  == 1                    ;DESTINATION
        000002                          XS.KND  == 2                    ;KNOWN DESTINATIONS
        000004                          XS.CTM  == 4                    ;COUNTER TIMER
        000010                          XS.MXC  == 10                   ;MAXIMUM CIRCUITS

        000020                          XS.CLM  == 20                   ;CALL MASK
        000040                          XS.CLV  == 40                   ;CALL VALUE
        000100                          XS.CUG  == 100                  ;GROUP
        000200                          XS.DTE  == 200                  ;NUMBER
        000400                          XS.OBJ  == 400                  ;OBJECT NUMBER
        001000                          XS.OBI  == 1000                 ;OBJECT ID
        002000                          XS.PRI  == 2000                 ;PRIORITY
        004000                          XS.SUB  == 4000                 ;SUBADDRESS RANGE
                                ;
                                ; MODULE OPTIONS - X25-ACCESS
                                ;
        000001                          XA.DST  == XS.DST               ;DESTINATION
        000002                          XA.KND  == XS.KND               ;KNOWN DESTINATIONS
        000004                          XA.NUM  == 2*XA.KND             ;NUMBER
                                ;
                                ; MODULE OPTIONS - X29 SERVER
                                ;
        000001                          X9.DST  == XS.DST               ;DESTINATION
        000002                          X9.KND  == XS.KND               ;KNOWN DESTINATIONS
        000004                          X9.CTM  == XS.CTM               ;COUNTER TIMER
        000010                          X9.MXC  == XS.MXC               ;MAXIMUM CIRCUITS

        000020                          X9.CLM  == XS.CLM               ;CALL MASK
        000040                          X9.CLV  == XS.CLV               ;CALL VALUE
        000100                          X9.CUG  == XS.CUG               ;GROUP
        000200                          X9.DTE  == XS.DTE               ;NUMBER
        000400                          X9.OBJ  == XS.OBJ               ;OBJECT NUMBER
        001000                          X9.OBI  == XS.OBI               ;OBJECT ID
        002000                          X9.PRI  == XS.PRI               ;PRIORITY
        004000                          X9.SUB  == XS.SUB               ;SUBADDRESS RANGE


                                ;
                                ; MODULE SNA-PROTOCOL OPTIONS
                                ;
        000001                          SP.ACC == 000001                ; ACCESS NAME
        000002                          SP.KNA == 000002                ; KNOWN ACCESS NAMES
        000004                          SP.CIR == 000004                ; CIRCUIT
        000010                          SP.SLU == 000010                ; SLU ADDRESS
        000020                          SP.PLU == 000020                ; PLU ADDRESS
        000040                          SP.MOD == 000040                ; MODE TABLE
        000100                          SP.UID == 000100                ; IBM USER
        000200                          SP.PSW == 000200                ; IBM PASSWORD
        000400                          SP.SPU == 000400                ; PU
        001000                          SP.KPU == 001000                ; KNOWN PUS
        002000                          SP.MLU == 002000                ; MAXIMUM LU
        004000                          SP.XID == 004000                ; LINK ID
```

```
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56      PAGE 4
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE              REFERENCES

ERRPRC    002400 RG        #13-1222
ERRSYN    002054 RG        #13-1214
EX$ERR  = 000002           #13-1212    13-1214     13-1216     13-1217     13-1218     13-1222     13-1223     13-1224     13-1225
                            13-1226
EX$SEV  = 000004           #13-1213    13-1215     13-1219     13-1220     13-1221
EX$SUC  = 000001           #13-1211
EX$WAR  = 000000           #13-1210
E2.BEA  = 000002  G        #11-860     11-873
E2.BRT  = 000001  G        #11-859     11-873
E2.INA  = 000010  G        #11-862     11-869
E2.INC  = 000020  G        #11-863     11-869
E2.NSP  = 000174  G        #11-869
E2.OUT  = 000040  G        #11-864     11-869
E2.RET  = 000100  G        #11-865     11-869
E2.ROU  = 000003  G        #11-873
E2.SEG  = 000004  G        #11-861     11-869
FD.CR   = ******  GX         6-105     6-111
FL.ADD  = 002000  G        #10-684
FL.CRE  = 001000  G        #10-682
FL.DDM  = 000001  G        #10-671
FL.DLC  = 000002  G        #10-672
FL.DLM  = 000200  G        #10-679
FL.KMX  = 000020  G        #10-675
FL.LLC  = 000004  G        #10-673
FL.LMC  = 000100  G        #10-677
FL.MFL  = 000010  G        #10-674
FL.MTP  = 000400  G        #10-680
FL.MUX  = 000040  G        #10-676
FL.NAM  = 004000  G        #10-686
FL.PVC  = 004000  G        #10-685
FL.SLV  = 001000  G        #10-681
FL.SNA  = 040000  G        #10-689
FL.TIM  = 000200  G        #10-678
FL.TRB  = 020000  G        #10-688
FL.TRI  = 010000  G        #10-687
FL.X25  = 002000  G        #10-683
FO.UPD  = ******  GX         6-113
FO.WRT  = ******  GX         6-106
F.DFNB  = 000046             6-95      6-106       6-113
F.DSPT  = 000044             6-95
F.FACC  = 000043             6-106     6-113
F.LUN   = 000042             6-95      6-106       6-113
F.NRBD  = 000024  G          6-100
F.RATT  = 000001             6-105     6-111
F.RTYP  = 000000             6-105     6-111
F.URBD  = 000020  G          6-95      6-95        6-95        6-95        6-100       6-112       6-112
G.MCRB  = 000002  G          6-95      6-95        6-95
IN$CHA  = 000004  G         #7-151
IN$COU  = 000006  G         #7-152
IN$EVE  = 000010  G         #7-153
IN$STA  = 000002  G         #7-150
IN$SUM  = 000000  G         #7-149
```

```
                                                                 K 3
CFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 17
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME        REFERENCES

ALUN$             #6-95     6-95
BDOFF$            #6-92
CBYTE$            #6-95     6-95      6-95     #6-95      6-95    #6-105     6-105     6-105    #6-106     6-106
                   6-106   #6-111     6-111     6-111    #6-112     6-112   #6-113     6-113     6-113
CSI$              #6-95     6-95
CWORD$            #6-95     6-95      6-95      6-95     #6-95      6-95      6-95     #6-95      6-95      6-95
                  #6-105    6-105     6-105     6-105    #6-106     6-106     6-106     6-106    #6-111     6-111
                   6-111    6-111    #6-112     6-112     6-112    #6-113     6-113     6-113     6-113    #6-115
                   6-115    6-115
DEF$L             #6-92
DIR$              #6-95
FDAT$A            #4-60     6-105     6-111
FDBDF$            #4-60    #6-95      6-95      6-104     6-110
FDBSZ$            #6-95     6-95     #6-104     6-104    #6-110     6-110
FDOFF$            #4-60    #6-95      6-95     #6-95      6-95      6-99     #6-105     6-105    #6-106     6-106
                  #6-111    6-111    #6-112     6-112    #6-113     6-113
FDOP$A            #4-60    #6-95      6-95      6-106     6-113
FDRC$A            #4-60    #6-95      6-95      6-112
FSRSZ$            #4-60     6-92
GCLDF$            #4-61     6-95
GCL$              #6-95
GMCR$             #6-95     6-95
NBOFF$            #6-95     6-95     #6-115     6-115
NMBLK$            #6-95     6-95      6-115
NSFDF$            #4-61     5-87
OFF$              #6-95    #6-95     #6-95     #6-95     #6-95     #7-127    #7-128
QDPB$             #6-95     6-95     #7-127     7-127    #7-128     7-128
QIOW$             #4-60     7-127     7-128
QIO$              #6-95     6-95
RAD50$            #6-95     6-95      6-95     #6-115     6-115     6-115
SETF$             #6-95     6-95
WTSE$             #6-95     6-95
```

L 3

```
                                             .SBTTL  $CFLIS - PROCESS LIST COMMAND
                                     ;+
                                     ; *** - $CFLIS - PROCESS LIST COMMAND
                                     ;
                                     ; INPUT:
                                     ;       $COMP = THE COMPONENT CODES
                                     ;
                                     ; OUTPUT:
                                     ;       CONTROL IS PASSED TO THE PROPER ROUTINE
                                     ;
                                     ;-

000152                               $CFLIS::
000152                                       DIR$    #GTIM                   ; GET CURRENT TIME
000160                                       DIR$    #TIDPB                  ; ATTACH TO TI:
000166   016700   000000G                    MOV     $COMP,R0                ; GET THE COMPONENT CODE
000172   006300                              ASL     R0                      ; CONVERT TO A WORD INDEX
000174   004770   177776'                    CALL    @DSPTBL-2(R0)           ; DISPATCH TO PROPER ROUTINE
000200                                       DIR$    #LFDPB                  ; PRINT A BLANK LINE
000206   012767   000000G 177622             MOV     #IO.DET,TIDPB+Q.IOFN ; SET TO DETACH FROM TI:
000214                                       DIR$    #TIDPB                  ; DETACH FROM TI:
000222   012767   000000G 177606             MOV     #IO.ATT,TIDPB+Q.IOFN ; SET TO ATTACH NEXT TIME
000230   000207                              RETURN


         000001                              .END
```

```
CFEOPN     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57     PAGE 2

MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CMOV$B             #5-107     5-107     5-107    #5-108    #5-108     5-108     5-108    #5-108     5-108    #7-181
                    7-181     7-181    #7-182    #7-182     7-182     7-182    #7-182     7-182
CMOV$W             #5-107     5-107     5-107     5-107    #5-108     5-108     5-108     5-108    #5-108     5-108
                    5-108    #6-137     6-137     6-137    #7-181     7-181     7-181     7-181    #7-182     7-182
                    7-182     7-182    #7-182     7-182     7-182    #8-202     8-202     8-202
CSI$               #4-62      4-69     #5-101    #5-103    #7-163    #7-165
CSI$1              #4-62      5-101     7-163
CSI$2              #4-62      5-103     7-165
DIR$               #4-63      5-114
ERR$               #5-108     5-108    #6-137     6-137    #7-182     7-182    #8-202     8-202
FDOFF$             #4-70      4-70
FDOF$L             #4-63      4-70
FDOP$R             #4-62      5-107    #5-108     5-108     7-181    #7-182     7-182
FDRC$R             #5-108     5-108    #7-182     7-182
GET$               #4-63      6-137
LDFDB$             #5-107     5-107    #5-108     5-108    #5-108     5-108    #6-137     6-137    #7-181     7-181
                   #7-182     7-182    #7-182     7-182    #8-202     8-202
LDRO$              #5-101     5-101    #5-103     5-103    #7-163     7-163    #7-165     7-165
NBOFF$             #4-71      4-71
NBOF$L             #4-63      4-71
OFF$               #4-73
OPEN$              #5-108     5-108    #7-182     7-182
OPEN$R             #4-63      5-108
OPEN$W             #4-62      7-182
PUT$               #4-62      8-202
QDPB$              #4-73      4-73
QIOW$              #4-63      4-73
```

```
001146  162767  000000C 000000G         SUB     #C$LENX+X$LENX,$BYTMN   ; UPDATE BYTE MINIMUM
001154  162767  000000C 000000G         SUB     #C$LENX+X$LENX,$BYTXT   ; UPDATE BYTE ALLOCATION
001162  000715                          BR      20$                     ; CONTINUE
                                ;
                                ; PURGE COUNTER TIMER
                                ;
001164  122762  000000G 000000G 50$:    CMPB    #CS.SVC,C.STS(R2) ; SVC?
001172  001406                          BEQ     60$                     ; BR IF YES
001174  026227  000000G 015355          CMP     P$VOWN(R2),#^RDLM ; DLM PVC ?
001202  001402                          BEQ     60$                     ; IF EQ, NOT THIS ONE
001204  005062  000000G                 CLR     P$VCT(R2)               ; CLEAR IT OUT
001210  000702                  60$:    BR      20$                     ; CONTINUE
                                ;
                                ; END OF LIST PROCESSING
                                ;
001212  005705                  70$:    TST     R5                      ; ELSE , ANYTHING FOUND?
001214  001406                          BEQ     102$                    ; BR IF ERROR
001216  000207                  80$:    RETURN
                                ;
                                ; ERRORS
                                ;
001220                          101$:   ERROR$  #ERRCTY                 ; ILLEGAL TYPE OF CIRCUIT
001230  000207                          RETURN
001232                          102$:   ERROR$  #ERRNSC                 ; CIRCUIT NOT IN CONFIGURATION
001242  000207                          RETURN
```

```
    282                                         ;
    283                                         ; COMMANDS
    284                                         ;
    285 000000                                          STATE$  PRSSTB,&P$CFE
    286 000000                                           TRAN$  &K$HEL, $EXIT,  .HLPAC, CM.HLP, $CMAND
    287 000000                                           TRAN$  &K$EXI, EXPUR,,         CM.EXI, $CMAND
    288 000000                                           TRAN$  &K$KIL, EXIT,,          CM.KIL, $CMAND
    289 000000                                           TRAN$  &K$DEF, DEFINE,,        CM.DEF, $CMAND
    290 000000                                           TRAN$  &K$SET, DEFINE,,        CM.DEF, $CMAND         ; SET = DEFINE
    291 000000                                           TRAN$  &K$PUR, PURGE,,         CM.PUR, $CMAND
    292 000000                                           TRAN$  &K$CLR, PURGE,,         CM.PUR, $CMAND         ; CLEAR = PURGE
    293 000000                                           TRAN$  &K$LIS, LIST,,          CM.LIS, $CMAND
    294 000000                                           TRAN$  &K$SHO, LIST,,          CM.LIS, $CMAND         ; SHOW = LIST
    295                                         ;
    296                                         ; EXIT OPTIONS
    297                                         ;
    298 000000                                          STATE$  EXPUR
    299 000000                                           TRAN$  &K$PUR, EXIT,,          SW.PUR, $SWICH
    300 000000                                           TRAN$  $EOS,EXIT
    929 000000                                  STATE$  RNSPA
    930 000000                                   TRAN$  &K$SER,RNSPA2
    931 000000                                   TRAN$  $LAMDA
    932 000000                                  STATE$  RNSPA2
    933 000000                                   TRAN$  &K$PAS
    934 000000                                  STATE$  ,&P$SERP
    935 000000                                   TRAN$  $STRNG, $EXIT,   .RNSPA
    936 000000                                   TRAN$  $LAMDA, $FAIL,   $CLERR, ERRILN, $ERROR
    937                                 ;
    938                                 ; Secondary [loader]
    939                                 ;
    940 000000                                  STATE$  RNSLO
    941 000000                                   TRAN$  &K$SEC
    942
    943 000000                                  STATE$
    944 000000                                   TRAN$  &K$LOAD,RNSLO2
    945 000000                                   TRAN$  $LAMDA
    946
    947 000000                                  STATE$  RNSLO2,&P$SLOF
    948 000000                                   TRAN$  $LAMDA,,         .RNCLF, $RNSLO, .RNFSP
    949 000000                                  STATE$
    950 000000                                   TRAN$  !FILSPC,$EXIT
    951                                 ;
    952                                 ; Tertiary [loader]
    953                                 ;
    954 000000                                  STATE$  RNTLO
    955 000000                                   TRAN$  &K$TER
    956
    957 000000                                  STATE$
    958 000000                                   TRAN$  &K$LOAD,RNTLO2
    959 000000                                   TRAN$  $LAMDA
    960
    961 000000                                  STATE$  RNTLO2,&P$TLOF
    962 000000                                   TRAN$  $LAMDA,,         .RNCLF, $RNTLO, .RNFSP
    963 000000                                  STATE$
    964 000000                                   TRAN$  !FILSPC,$EXIT
    965                                 ;
    966                                 ; Load file ( Load [file] or [Load] file)
    967                                 ;
    968 000000                                  STATE$  RNLOD
    969 000000                                   TRAN$  &K$LOAD,RNLOD2
    970 000000                                   TRAN$  $LAMDA, RNLOD3
    971
    972 000000                                  STATE$  RNLOD2
    973 000000                                   TRAN$  &K$FIL, RNLOD4
    974 000000                                   TRAN$  $LAMDA, RNLOD4
    975
    976 000000                                  STATE$  RNLOD3
    977 000000                                   TRAN$  &K$FIL
    978
    979 000000                                  STATE$  RNLOD4,&P$LOF
    980 000000                                   TRAN$  $LAMDA,,         .RNCLF, $RNLOD, .RNFSP
    981 000000                                  STATE$
    982 000000                                   TRAN$  !FILSPC,$EXIT
    983                                 ;
    984                                 ; Diagnostics [file]
    985                                 ;
```

```
   1590                                              .SBTTL  DEFINE EXECUTOR COMMAND
   1591                                      ;
   1592                                      ; DEFINE EXECUTOR COMMAND
   1593                                      ;
   1594 000000                                       STATE$  DEFEXE
   1595 000000                                        TRAN$  &K$EXE,,,CO.EXE,$COMP
   1596
   1597 000000                                       STATE$  ,&P$EXE
   1598 000000                                        TRAN$  !EXEOPT
   1599
   1600 000000                                       STATE$  EXEO
   1601 000000                                        TRAN$  $EOS,$EXIT
   1602 000000                                        TRAN$  !EXEOPT,EXEO
   1603                                      ;
   1604                                      ; LOOK FOR EXECUTOR OPTIONS
   1605                                      ;
   1606 000000                                       STATE$  EXEOPT
   1607 000000                                        TRAN$  !ENAME, $EXIT,, OE.NAM, $OPRND
   1608 000000                                        TRAN$  !EADD,  $EXIT,, OE.ADD, $OPRND
   1609 000000                                        TRAN$  !ELINK, $EXIT,, OE.LNK, $OPRND
   1610 000000                                        TRAN$  !EID,   $EXIT,, OE.ID,  $OPRND
   1611 000000                                        TRAN$  !EVFY,  $EXIT,, OE.VFY, $OPRND
   1612 000000                                        TRAN$  !EHOP,  $EXIT,, OE.MXH, $OPRND
   1613 000000                                        TRAN$  !ECOST, $EXIT,, OE.MXC, $OPRND
   1614 000000                                        TRAN$  !ENN,   $EXIT,, OE.NN,  $OPRND
   1615 000000                                        TRAN$  !EHOST, $EXIT,, OE.HOS, $OPRND
   1616 000000                                        TRAN$  !ENCOU, $EXIT,, OE.NCO, $OPRND
   1617 000000                                        TRAN$  !EROU,  $EXIT,, OE.RTM, $OPRND
   1618 000000                                        TRAN$  !ESUB,  $EXIT,, OE.SUB, $OPRND
   1619 000000                                        TRAN$  !EAMAX, $EXIT,, OE.AMAX,$OPRND
   1620 000000                                        TRAN$  !EAHOPS,$EXIT,, OE.AHOP,$OPRND
   1621 000000                                        TRAN$  !EACOST,$EXIT,, OE.ACOS,$OPRND
   1622 000000                                        TRAN$  !EBRT,  $EXIT,, E2.BRT, $OPRN2
   1623 000000                                        TRAN$  !EBEA,  $EXIT,, E2.BEA, $OPRN2
   1624 000000                                        TRAN$  !ESEG,  $EXIT,, E2.SEG, $OPRN2
   1625 000000                                        TRAN$  !EINAC, $EXIT,, E2.INA, $OPRN2
   1626 000000                                        TRAN$  !EINCT, $EXIT,, E2.INC, $OPRN2
   1627 000000                                        TRAN$  !EOUTT, $EXIT,, E2.OUT, $OPRN2
   1628 000000                                        TRAN$  !ERETF, $EXIT,, E2.RET, $OPRN2
   1629                                      ;
   1630                                      ; LOOK FOR NODE NAME
   1631                                      ;
   1632 000000                                       STATE$  ENAME
   1633 000000                                        TRAN$  &K$NAM
   1634
   1635 000000                                       STATE$  ,&P$NAM
   1636 000000                                        TRAN$  $RAD50, $EXIT,   .EXNAM
   1637                                      ;
   1638                                      ; LOOK FOR NODE ADDRESS
   1639                                      ;
   1640 000000                                       STATE$  EADD
   1641 000000                                        TRAN$  &K$ADD
   1642
   1643 000000                                       STATE$  ,&P$ADD
   1644 000000                                        TRAN$  !NODADD,$EXIT,   .EXADD
   1645                                      ;
   1646                                      ; LOOK FOR HOST ADDRESS
```

```
000000                           TRAN$  !CHNL,  $EXIT,, XP.CHN, $SWICH

                         ;
                         ; X25-PROTOCOL GROUP PARAMETERS
                         ;
000000                          STATE$  DXPCUG,&P$XPRC
000000                           TRAN$  !KNG,   DXPCG0,,XP.KNG, $OPRND
000000                           TRAN$  !CUG,         ,,XP.CUG, $OPRND
000000                          STATE$  DXPCG0, &P$GRP0
000000                           TRAN$  !DXPCG2
000000                          STATE$  DXPCG1
000000                           TRAN$  $EOS,   $EXIT
000000                           TRAN$  !DXPCG2,DXPCG1

000000                          STATE$  DXPCG2
000000                           TRAN$  !DTE,   $EXIT,, XP.DTA, $SWICH
000000                           TRAN$  !GNUM,  $EXIT,, XP.NUM, $SWICH
000000                           TRAN$  !TBIL,  $EXIT,, XP.BIL, $SWICH

                         ;
                         ; SUBEXPRESSION TO MATCH CALL TIMER
                         ;
000000                          STATE$  CTM
000000                           TRAN$  &K$CALL
000000                          STATE$  ,       &P$CALL
000000                           TRAN$  &K$TIM
000000                          STATE$  ,       &P$CALT
000000                           TRAN$  $DNUMB, $EXIT,   .MOCAT
                         ;
                         ; SUBEXPRESSION TO MATCH CLEAR TIMER
                         ;
000000                          STATE$  CLT
000000                           TRAN$  &K$CLR
000000                          STATE$  ,       &P$CLRO
000000                           TRAN$  &K$TIM
000000                          STATE$  ,       &P$CLRT
000000                           TRAN$  $DNUMB, $EXIT,   .MOCLT
                         ;
                         ; SUBEXPRESSION TO PARSE DEFAULT DATA
                         ;
000000                          STATE$  DFB
000000                           TRAN$  &K$DFA
000000                          STATE$  ,       &P$DFAO
000000                           TRAN$  &K$DAT
000000                          STATE$  ,       &P$DFDA
000000                           TRAN$  $DNUMB, $EXIT,   .MODFD
                         ;
                         ; SUBEXPRESSION TO PARSE DEFAULT WINDOW
                         ;
000000                          STATE$  DFW
000000                           TRAN$  &K$DFA
000000                          STATE$  ,       &P$DFAO
000000                           TRAN$  &K$WIN
000000                          STATE$  ,       &P$DFWI
000000                           TRAN$  $DNUMB, $EXIT,   .MODFW
                         ;
                         ; SUBEXPRESSION TO PARSE MAXIMUM DATA
```

```
                                                                     K 11
CFEPRS     CREATED BY  MACRO  ON 2-JUL-85 AT 14:24       PAGE 1
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE             REFERENCES

PRSKTB    000000 RG       #4-84
PRSSTB    000000 RG       #4-84
QUOTE   = 000042          #4-76
RTBRK   = 000076          #4-77
SF.PH3  = 000400          #7-867
$ALPHA  = 000022          #4-84
$ANY    = 000020          #4-84
$BLANK  = 000006          #4-84
$DIGIT  = 000024          #4-84
$DNUMB  = 000014          #4-84
$EOS    = 000012          #4-84
$EXIT   = 000000          #4-84
$FAIL   = 177777          #4-84
$GOFF   = 000001          #4-69     4-84
$GPRM   = 000001          #4-65     4-84
$LAMDA  = 000000          #4-84
$NUMBR  = 000002          #4-84
$RAD50  = 000016          #4-84
$RONLY  = 000001          #4-73     4-84      4-84      4-84      4-84
$STRNG  = 000004          #4-84
$SUBXP  = 000010          #4-84
$$$FLG  = 177777          #4-84
$$$KEY  = 177777          #4-84
```

L 11

```
CFEROT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:13     PAGE 1
SYMBOL  CROSS REFERENCE                                 CREF    04.00

SYMBOL  VALUE              REFERENCES

ERRCON  = ******   GX       5-158
EX$ERR  = 000002           #5-165
EX$SEV  = 000004           #5-165
EX$SUC  = 000001           #5-165
EX$WAR  = 000000           #5-165
F.FNB   = 000102            4-103       5-145      *5-147
F.LUN   = 000042            5-151
IOSB      000030 R          4-103      #4-104
IO.WAT  = ******   GX       4-103
N.FID   = 000000            4-103       5-145      *5-147
PAR$$$  = 000027           #5-143       5-143      #5-146       5-146      #5-163       5-163      #5-164       5-164
Q.IOFN  = 000002            4-103
Q.IOLU  = 000004           *5-151
SW.PUR  = ******   GX       5-141
WATDPB    000000 R         #4-103      *5-151       5-152
$CFCOM  = ******   GX       5-131
$CFDSP  = ******   GX       5-133
$CFE      000034 RG        #5-128       5-170
$CFERR  = ******   GX       5-158
$CFINI  = ******   GX       5-129
$CFUPD  = ******   GX       5-139
$EXSTS  = ******   GX       5-165
$FIATR  = ******   GX       4-103       5-148      *5-150
$FIN      000056 RG         5-132      #5-138
$IFDB   = ******   GX       5-143       5-143       5-164       5-164
$OFDB   = ******   GX       4-103       5-144       5-146       5-146       5-163       5-163
$SWICH  = ******   GX       5-141
$$$ARG  = 000002           #4-103       4-103      #4-103       4-103      #4-103       4-103       4-103
$$$GLB  = ******            4-103       5-165       5-165       5-165       5-165
$$$OST  = 000005           #5-165       5-165       5-165      #5-165       5-165       5-165      #5-165       5-165       5-165
                           #5-165       5-165       5-165      #5-165
.CLOSE  = ******    G       5-146       5-146       5-164       5-164
.CONER    000162 RG        #5-157
.DELET  = ******    G       5-143       5-143       5-163       5-163
.EXIT     000172 RG         5-140      #5-162
```

L 12

```
**FILE**ID**CFESTB




  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE     SSSSSSSS   TTTTTTTTTT   BBBBBBBB
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE     SSSSSSSS   TTTTTTTTTT   BBBBBBBB
CC           FF           EE           SS               TT       BB      BB
CC           FF           EE           SS               TT       BB      BB
CC           FF           EE           SS               TT       BB      BB
CC           FF           EE           SS               TT       BB      BB
CC           FFFFFFFF     EEEEEEEE       SSSSSS         TT       BBBBBBBB
CC           FFFFFFFF     EEEEEEEE       SSSSSS         TT       BBBBBBBB
CC           FF           EE                   SS       TT       BB      BB
CC           FF           EE                   SS       TT       BB      BB
CC           FF           EE                   SS       TT       BB      BB      ....
CC           FF           EE                   SS       TT       BB      BB      ....
  CCCCCCCC   FF           EEEEEEEEEE   SSSSSSSS         TT       BBBBBBBB        ....
  CCCCCCCC   FF           EEEEEEEEEE   SSSSSSSS         TT       BBBBBBBB        ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                .TITLE  CFESUB - COMMON SUBROUTINES
                .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       COMMON UTILITY ROUTINES
;
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  10-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                 .SBTTL  $FNSLT - FIND AN SLT TEMPLATE
                                        ;+
                                        ; *** - $FNSLT - FIND AN SLT TEMPLATE
                                        ;
                                        ; INPUTS:
                                        ;        R5 = POINTER TO CURRENT TEMPLATE
                                        ;        $DVNAM = START OF LINE DESCRIPTOR
                                        ;
                                        ; OUTPUTS:
                                        ;        C-BIT = SUCCESS/FAILURE
                                        ;        R5 = ADDRESS OF TEMPLATE
                                        ;        $FLAGS = UPDATED
                                        ;
                                        ;-

000726  005727                          .FNSLT::TST     (PC)+                   ; SPECIAL FOR SET KNOWN LINES
000730  000261                          $FNSLT::SEC                             ; ENTRY POINT FOR OLD STUFF
000732  006046                                  ROR     -(SP)                   ; PUSH ENTRY POINT FLAG
000734  012701  000000G                         MOV     #$DVNAM,R1              ; POINT TO LINE DESCRIPTOR
000740  011505                          10$:    MOV     (R5),R5                 ; GET NEXT TEMPLATE
000742  000261                                  SEC                             ; ASSUME FAILURE
000744  001510                                  BEQ     20$                     ; IF EQ, END OF LIST
000746  122765  000000G 000000G                 CMPB    #CS.SLT,C.STS(R5)       ; IS THIS AN SLT TEMPLATE ?
000754  001371                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
000756  005716                                  TST     (SP)                    ; SPECIAL HANDLING FOR KNOWN LINES ?
000760  100404                                  BMI     11$                     ; IF MI, NO
000762  032767  000000G 000000G                 BIT     #WC.NAM,$WILD           ; ELSE, IS IT SET KNOWN LINES ?
000770  001004                                  BNE     12$                     ; IF NE, YES
000772  026165  000000G 000004  11$:            CMP     L$NAM(R1),C.NAM(R5)     ; IS THIS THE RIGHT SLT ?
001000  001357                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
001002  042767  000000G 000000G 12$:            BIC     #FL.MTP,$FLAGS          ; ASSUME LINE IS NOT MULTI-POINT
001010  032765  000000G 000006                  BIT     #LF.MTP,C.FLG(R5)       ; IS THE LINE MULTIPOINT ?
001016  001403                                  BEQ     13$                     ; IF EQ, NO
001020  052767  000000G 000000G                 BIS     #FL.MTP,$FLAGS          ; ELSE, REMEMBER LINE IS MULTIPOINT
001026  042767  000000G 000000G 13$:            BIC     #FL.MFL,$FLAGS          ; ASSUME LINE IS NOT MARKED FOR LOAD
001034  032765  000000G 000006                  BIT     #LF.MFL,C.FLG(R5)       ; IS THE LINE MARKED FOR LOAD ?
001042  001403                                  BEQ     14$                     ; IF EQ, NO
001044  052767  000000G 000000G                 BIS     #FL.MFL,$FLAGS          ; ELSE, REMEMBER LINE IS MARKED FOR LOAD
001052  032767  000000G 000000G 14$:            BIT     #WC.CNT,$WILD           ; ALL CONTROLLERS ?
001060  001041                                  BNE     17$                     ; IF NE, YES
001062  126165  000000G 000010                  CMPB    L$CNT(R1),C.CNT(R5)     ; IS IT THE RIGHT CONTROLLER ?
001070  001323                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
001072  032767  000000G 000000G                 BIT     #FL.MUX,$FLAGS          ; IS THIS DEVICE A MULTIPLEXER ?
001100  001410                                  BEQ     15$                     ; IF EQ, NO
001102  032767  000000G 000000G                 BIT     #WC.UNT,$WILD           ; ALL UNITS ?
001110  001025                                  BNE     17$                     ; IF NE, YES
001112  126165  000000G 000011                  CMPB    L$UNT(R1),C.UNT(R5)     ; IS IT THE RIGHT UNIT ?
001120  001307                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
001122  032765  000000G 000006  15$:            BIT     #LF.MTP,C.FLG(R5)       ; IS THE LINE MULTI-POINT ?
001130  001415                                  BEQ     17$                     ; IF EQ, NO
001132  032767  000000G 000000G                 BIT     #FL.TRI,$FLAGS          ; DID HE WANT THE TRIBUTARY INFO ?
001140  001411                                  BEQ     17$                     ; IF EQ, NO
001142  032767  000000G 000000G                 BIT     #WC.TRI,$WILD           ; ALL TRIBUTARIES ?
001150  001005                                  BNE     17$                     ; IF NE, YES
001152  010546                                  MOV     R5,-(SP)                ; SAVE R5
001154  004767  000012                          CALL    $FNSTA                  ; ELSE, TRY TO FIND THE STATION
001160  012605                                  MOV     (SP)+,R5                ; RESTORE R5
```

```
CFESUB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:14      PAGE 6
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

CTRDF$           #4-61        4-63
CUGDF$           #5-98        5-98
DECTP$           #4-61        4-64
DIR$             #4-61        7-173       8-214
DSTDF$           #5-101       5-101
DTEDF$           #5-96        5-96
FLTDF$           #5-95        5-95
LLTDF$           #5-111       5-111
OBJDF$           #5-93        5-93
PVCDF$           #5-97        5-97
RDTDF$           #5-99        5-99
REDEF            #5-83        5-93        5-94       5-95       5-96       5-97       5-98       5-99       5-100      5-101
                  5-102       5-103       5-111
RNBDF$           #5-94        5-94
SERDF$           #5-103       5-103
SVCDF$           #5-102       5-102
XCBDF$           #5-100       5-100
```

```
   1033         010000                            SP.SID == 010000                ; SSCP ID
   1034         020000                            SP.DAT == 020000                ; USER DATA
```

```
                                                                          L  2
LFEIMP     CREATED BY  MACRO  ON 29-JUN-85 AT 04:56     PAGE 5
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE              REFERENCES

IO.WVB  = ******  GX        7-127       7-128
LF.BRO  = 000100  G        #12-1077
LF.ENA  = 000001  G        #12-1071
LF.MDC  = 000002  G        #12-1072
LF.MFL  = 000010  G        #12-1074
LF.MTP  = 000004  G        #12-1073
LF.SER  = 000200  G        #12-1078
LF.TIM  = 000020  G        #12-1075
LF.X2P  = 000040  G        #12-1076
LUNX    = 000006           #5-73
LUN1    = 000001           #5-68        6-95        6-95        6-95        6-95
LUN2    = 000002           #5-69        7-127       7-127
LUN3    = 000003           #5-70        7-128       7-128
LUN4    = 000004           #5-71        6-106
LUN5    = 000005           #5-72        6-113
L$CNT     000002  G        #8-338
L$LCTM    000016  G        #9-451
L$LDEV    000012  G        #9-448
L$LEXT    000015  G        #9-450
L$LFLG    000006  G        #9-445
L$LLEN  = 000020  G        #9-452
L$LLIN    000011  G        #9-447
L$LNAM    000004  G        #9-444
L$LNLT    000014  G        #9-449
L$LPRI    000010  G        #9-446
L$NAM     000000  G        #8-337
L$TRI     000006  G        #8-340
L$UNT     000004  G        #8-339
MAXARE  = 000077  G        #5-77
MAXCST  = 001776  G        #5-76
MAXHOP  = 000036  G        #5-75
MAXNN   = 001777  G        #5-78
MXCSTL  = 000031  G        #5-79
NF$ACC  = 100000  G        #5-87
NF$EVT  = 040000  G        #5-87
NF.ACC  = 000001  G        #5-87
NF.BCP  = 000040  G        #5-87
NF.END  = 000200  G        #5-87
NF.EVT  = 000002  G        #5-87
NF.LLI  = 000004  G        #5-87
NF.LV2  = 000100  G        #5-87
NF.SLI  = 000400  G        #5-87
NF.SMC  = 000020  G        #5-87
NF.SWP  = 000010  G        #5-87
NO$EXA  = 000400  G        #12-1181
NO$HOA  = 001000  G        #12-1182
N.DVNM  = 000032             6-95       6-115
N.FNAM  = 000006             6-95       6-115
N.FTYP  = 000014             6-95       6-115
N.UNIT  = 000034             6-95       6-115
OC.APR  = 010000  G        #11-807      11-821
OC.CHN  = 000040  G        #11-798      11-816      11-823
```

M 2

```
**FILE**ID**CFEINI




  CCCCCCCC   FFFFFFFFFF    EEEEEEEEEE     IIIIII     NN       NN     IIIIII
  CCCCCCCC   FFFFFFFFFF    EEEEEEEEEE     IIIIII     NN       NN     IIIIII
CC           FF            EE               II       NN       NN       II
CC           FF            EE               II       NN       NN       II
CC           FF            EE               II       NNNN     NN       II
CC           FF            EE               II       NNNN     NN       II
CC           FFFFFFFF      EEEEEEEE         II       NN   NN  NN       II
CC           FFFFFFFF      EEEEEEEE         II       NN   NN  NN       II
CC           FF            EE               II       NN     NNNN       II
CC           FF            EE               II       NN     NNNN       II
CC           FF            EE               II       NN       NN       II            ....
CC           FF            EE               II       NN       NN       II            ....
  CCCCCCCC   FF            EEEEEEEEEE     IIIIII     NN       NN     IIIIII          ....
  CCCCCCCC   FF            EEEEEEEEEE     IIIIII     NN       NN     IIIIII          ....




LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                             TT
LL                             TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
CFELIS - PERFORM LIST COMMAND   MACRO V05.03b Saturday 29-Jun-85 04:57  Page 5-1
Symbol table

BLANK   000150R         G.TIMO= 000002          Q.IOFN= 000002          $HDRBK  000120RG        $SHSNA= ****** GX
DSPTBL  000000R         G.TISC= 000012          Q.IOLU= 000004          $SHCIR= ****** GX       $SHSYS= ****** GX
GTIM    000114R         G.TIYR= 000000          Q.IOPL= 000014          $SHEXE= ****** GX       $SHXAC= ****** GX
G.TIBA= 000002          IO.ATT= ****** GX       Q.IOPR= 000007          $SHLIN= ****** GX       $SHXPR= ****** GX
G.TICP= 000016          IO.DET= ****** GX       Q.IOSB= 000010          $SHLOG= ****** GX       $SHXSV= ****** GX
G.TICT= 000014          IO.WVB= ****** GX       TIDPB   000034R         $SHOBJ= ****** GX       $SHX29= ****** GX
G.TIDA= 000004          LFDPB   000064R         $CFLIS  000152RG        $SHPRO= ****** GX       $$$ARG= 000003
G.TIHR= 000006          Q.IOAE= 000012          $COMP = ****** GX       $SHREM= ****** GX       $$$OST= 000020
G.TIMI= 000010          Q.IOEF= 000006          $GTBUF  000130RG

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000232    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8800 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.28
SY:CFELIS.V2,[132,134]CFELIS/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFELIS
```

```
**FILE**ID**CFEPRG
```

```
  CCCCCCCC    FFFFFFFFFF    EEEEEEEEEE    PPPPPPPP      RRRRRRRR        GGGGGGGG
  CCCCCCCC    FFFFFFFFFF    EEEEEEEEEE    PPPPPPPP      RRRRRRRR        GGGGGGGG
CC            FF            EE            PP      PP    RR      RR    GG
CC            FF            EE            PP      PP    RR      RR    GG
CC            FF            EE            PP      PP    RR      RR    GG
CC            FF            EE            PP      PP    RR      RR    GG
CC            FFFFFFFF      EEEEEEEE      PPPPPPPP      RRRRRRRR      GG
CC            FFFFFFFF      EEEEEEEE      PPPPPPPP      RRRRRRRR      GG
CC            FF            EE            PP            RR  RR        GG  GGGGGG
CC            FF            EE            PP            RR  RR        GG  GGGGGG
CC            FF            EE            PP            RR    RR      GG      GG      ....
CC            FF            EE            PP            RR    RR      GG      GG      ....
  CCCCCCCC    FF            EEEEEEEEEE    PP            RR      RR      GGGGGG        ....
  CCCCCCCC    FF            EEEEEEEEEE    PP            RR      RR      GGGGGG        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

```
                                                .SBTTL  QREM - REMOVE ENTRY FROM QUEUE
                                        ;+
                                        ; *** - QREM - REMOVE ENTRY FROM QUEUE
                                        ;
                                        ; INPUTS:
                                        ;       R5 = PREVIOUS ENTRY IN QUEUE
                                        ;       R2 = ENTRY TO REMOVE
                                        ;       $TMLST = QUEUE LISTHEAD
                                        ;
                                        ; OUTPUTS:
                                        ;       R5,R2 = PRESERVED
                                        ;       ENTRY IS REMOVED FROM QUEUE
                                        ;
                                        ;-

 001244                                 QREM:
 001244 011215                                  MOV     (R2),(R5)       ; CLOSE UP LIST
 001246 001002                                  BNE     10$             ; IF NE, NOT LAST IN LIST
 001250 010567  000002G                         MOV     R5,$TMLST+2     ; SET NEW LAST ENTRY
 001254 000207                          10$:    RETURN


        000001                                  .END
```

```
                                        .SBTTL  LIST COMPONENTS
                               ;
                               ; LIST COMPONENTS
                               ;
000000                                  STATE$  LIST,&P$LIS
000000                                   TRAN$  !LISMOD,EXIT
000000                                   TRAN$  !LISOPT,LISCM1
000000                                   TRAN$  !LISCOM,LISCMO
000000                                   TRAN$  !LISLOG

000000                                  STATE$
000000                                   TRAN$  &K$EVE, EXIT
000000                                   TRAN$  $LAMDA
000000                                  STATE$  LISCMO
000000                                   TRAN$  !LISOPT,EXIT
000000                                   TRAN$  $LAMDA, EXIT

000000                                  STATE$  LISCM1
000000                                   TRAN$  !LISCOM,EXIT

000000                                  STATE$  LISCOM, &P$LIS
000000                                   TRAN$  !KNLIN, $EXIT,,         CO.KNL, $COMP
000000                                   TRAN$  !LINID, $EXIT,,         CO.LIN, $COMP
000000                                   TRAN$  !CIRID, $EXIT,,         CO.CIR, $COMP
000000                                   TRAN$  !KNCIR, $EXIT,,         CO.KNC, $COMP
000000                                   TRAN$  &K$EXE, $EXIT,,         CO.EXE, $COMP
000000                                   TRAN$  !KNPRO, $EXIT,,         CO.KNP, $COMP
000000                                   TRAN$  !PROID, $EXIT,,         CO.PRO, $COMP
000000                                   TRAN$  &K$SYS, $EXIT,,         CO.SYS, $COMP
000000                                   TRAN$  !KNOBJ, $EXIT,,         CO.KNO, $COMP
000000                                   TRAN$  !OBJID, $EXIT,,         CO.OBJ, $COMP
000000                                   TRAN$  !NODID, $EXIT,,         CO.NOD, $COMP
000000                                   TRAN$  !KNNOD, $EXIT,,         CO.KNN, $COMP

000000                                  STATE$  LISLOG
000000                                   TRAN$  !LOGID, $EXIT,,         CO.LOG, $COMP
000000                                   TRAN$  !KNLOG, $EXIT,,         CO.KLG, $COMP

                               ;
                               ; LIST COMPONENTS OPTIONS
                               ;
000000                                  STATE$  LISOPT
000000                                   TRAN$  $EOS,    $EXIT
000000                                   TRAN$  &K$CHA, $EXIT,,         IN$CHA, $INFO
000000                                   TRAN$  &K$SUM, $EXIT,,         IN$SUM, $INFO
000000                                   TRAN$  &K$STS, $EXIT,,         IN$SUM, $INFO
                               ;
                               ; LIST MODULES
                               ;
000000                                  STATE$  LISMOD
000000                                   TRAN$  !MODXPR,LIXPR,,         CO.MXP, $COMP
000000                                   TRAN$  !LXPRAB,LISOPT,,        CO.MXP, $COMP
000000                                   TRAN$  !MODXSV,LIXSR,,         CO.MXS, $COMP
000000                                   TRAN$  !MODX29,LIXSR,,         CO.MX9, $COMP
000000                                   TRAN$  !MODXAC,LIXAC,,         CO.MAC, $COMP
000000                                   TRAN$  !LXSRAB,LISOPT,,        CO.MXS, $COMP
000000                                   TRAN$  !MODSPR,$EXIT,$SNLIS,CO.MSP,$COMP ; LIST MOD SNA-PROTOCOL
```

```
    986 000000                              STATE$  RNDIA
    987 000000                               TRAN$  &K$DIAG
    988
    989 000000                              STATE$
    990 000000                               TRAN$  &K$FIL,RNDIA2
    991 000000                               TRAN$  $LAMDA
    992
    993 000000                              STATE$  RNDIA2,&P$DIAF
    994 000000                               TRAN$  $LAMDA,,        .RNCLF, $RNDIA, .RNFSP
    995 000000                              STATE$
    996 000000                               TRAN$  !FILSPC,$EXIT
    997                              ;
    998                              ; Dump file
    999                              ;
   1000 000000                              STATE$  RNDMP
   1001 000000                               TRAN$  &K$DMP
   1002
   1003 000000                              STATE$  ,&P$DMPO
   1004 000000                               TRAN$  &K$FIL,RNDMP2
   1005
   1006 000000                              STATE$  RNDMP2,&P$DMPF
   1007 000000                               TRAN$  $LAMDA,,        .RNCLF, $RNDMP, .RNFSP
   1008 000000                              STATE$
   1009 000000                               TRAN$  !FILSPC,$EXIT
   1010
   1011                              ;
   1012                              ; Subexpression to parse filespec
   1013                              ;
   1014 000000                              STATE$  FILSPC
   1015 000000                               TRAN$  !STRNG,$EXIT,   CKFSP
   1016 000000                               TRAN$  $LAMDA, $FAIL,  $CLERR, ERRFSP, $ERROR
   1017
   1018                              ;
   1019                              ; NODE ID
   1020                              ;
   1021 000000                              STATE$  NODID
   1022 000000                               TRAN$  &K$NOD
   1023
   1024 000000                              STATE$  ,&P$NID
   1025 000000                               TRAN$  !NODADD,NODID3
   1026 000000                               TRAN$  $LAMDA, NODID1
   1027
   1028 000000                              STATE$  NODID3
   1029 000000                               TRAN$  $LAMDA, $EXIT,  .RNAD3
   1030
   1031 000000                              STATE$  NODID1
   1032 000000                               TRAN$  $STRNG,,        .RNNAM
   1033
   1034 000000                              STATE$
   1035 000000                               TRAN$  <':>,   NODID2
   1036 000000                               TRAN$  $LAMDA, $EXIT
   1037
   1038 000000                              STATE$  NODID2
   1039 000000                               TRAN$  <':>,   $EXIT
   1040                              ;
   1041                              ; NODE ADDRESS
   1042                              ;
```

```
                      ;
000000                        STATE$  EHOST
000000                         TRAN$  &K$HOS

000000                        STATE$  ,&P$HOS
000000                         TRAN$  $LAMDA,,           .EXH01
000000                        STATE$
000000                         TRAN$  !NODADD,$EXIT,     .EXHOS
000000                         TRAN$  !HOSNAM,$EXIT,     .EXHOS
000000                         TRAN$  $LAMDA, $FAIL

                      ;
                      ; LOOK FOR LINKS
                      ;
000000                        STATE$  ELINK
000000                         TRAN$  &K$MAX

000000                        STATE$  ,&P$PEN
000000                         TRAN$  &K$LNKS

000000                        STATE$  ,&P$MLNK
000000                         TRAN$  $DNUMB, $EXIT,     .EXLNK
                      ;
                      ; LOOK FOR NODE ID
                      ;
000000                        STATE$  EID
000000                         TRAN$  &K$IDE, EID1
000000                         TRAN$  &K$ID

000000                        STATE$  EID1
000000                         TRAN$  !QSTRNG,$EXIT,     .EXID
000000                         TRAN$  !STRNG, $EXIT,     .EXID1
                      ;
                      ; NON QUOTED STRING
                      ;
000000                        STATE$  STRNG
000000                         TRAN$  $LAMDA,,           $BL.MAN

000000                        STATE$  STRING
000000                         TRAN$  $BLANK, $EXIT,     $BL.AUT
000000                         TRAN$  $EOS,   $EXIT,     $BL.AUT
000000                         TRAN$  $ANY,   STRING
                      ;
                      ; QUOTED STRING
                      ;
000000                        STATE$  QSTRNG
000000                         TRAN$  QUOTE,,            $BL.MAN

000000                        STATE$  STRNG1
000000                         TRAN$  !DQUOTE,STRNG1
000000                         TRAN$  QUOTE,  $EXIT,     $BL.AUT
000000                         TRAN$  $ANY,   STRNG1

000000                        STATE$  DQUOTE
000000                         TRAN$  QUOTE
000000                        STATE$
000000                         TRAN$  QUOTE,  $EXIT
```

```
  2308                                      ;
  2309 000000                                        STATE$  MMDA
  2310 000000                                         TRAN$  &K$MAX
  2311 000000                                        STATE$  ,           &P$MAX
  2312 000000                                         TRAN$  &K$DAT
  2313 000000                                        STATE$  ,           &P$MBD
  2314 000000                                         TRAN$  $DNUMB,  $EXIT,   .MOMDA
  2315
  2316                                      ;
  2317                                      ;   SUBEXPRESSION TO PARSE MAXIMUM CLEARS
  2318                                      ;
  2319 000000                                        STATE$  MXC
  2320 000000                                         TRAN$  &K$MAX
  2321 000000                                        STATE$  ,           &P$PEN
  2322 000000                                         TRAN$  &K$CLRS
  2323 000000                                        STATE$  ,           &P$MCLR
  2324 000000                                         TRAN$  $DNUMB,  $EXIT,   .MOMCL
  2325                                      ;
  2326                                      ;   SUBEXPRESSION TO PARSE MAXIMUM RESETS
  2327                                      ;
  2328 000000                                        STATE$  MRE
  2329 000000                                         TRAN$  &K$MAX
  2330 000000                                        STATE$  ,           &P$PEN
  2331 000000                                         TRAN$  &K$RES
  2332 000000                                        STATE$  ,           &P$MRES
  2333 000000                                         TRAN$  $DNUMB,  $EXIT,   .MOMRE
  2334                                      ;
  2335                                      ;   SUBEXPRESSION TO PARSE RESET TIMER
  2336                                      ;
  2337 000000                                        STATE$  RET
  2338 000000                                         TRAN$  &K$RES
  2339 000000                                        STATE$  ,           &P$RESO
  2340 000000                                         TRAN$  &K$TIM
  2341 000000                                        STATE$  ,           &P$REST
  2342 000000                                         TRAN$  $DNUMB,  $EXIT,   .MORET
  2343                                      ;
  2344                                      ;   SUBEXPRESSION TO PARSE MAXIMUM RESTARTS
  2345                                      ;
  2346 000000                                        STATE$  MXR
  2347 000000                                         TRAN$  &K$MAX
  2348 000000                                        STATE$  ,           &P$PEN
  2349 000000                                         TRAN$  &K$RSTR
  2350 000000                                        STATE$  ,           &P$MRST
  2351 000000                                         TRAN$  $DNUMB,  $EXIT,   .MOMXR
  2352                                      ;
  2353                                      ;   SUBEXPRSESSION TO PARSE RESTART TIMER
  2354                                      ;
  2355 000000                                        STATE$  RST
  2356 000000                                         TRAN$  &K$RSTR
  2357 000000                                        STATE$  ,           &P$RSTO
  2358 000000                                         TRAN$  &K$TIM
  2359 000000                                        STATE$  ,           &P$RSTT
  2360 000000                                         TRAN$  $DNUMB,  $EXIT,   .MORST
  2361
  2362                                      ;
  2363                                      ;   SUBEXPRESSION TO PARSE NETWORK NAME
  2364                                      ;
```

```
CFEPRS     CREATED BY  MACRO  ON 2-JUL-85 AT 14:24      PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME     REFERENCES

DBGTP$         #4-84     #4-287    #4-288    #4-289    #4-290    #4-291    #4-292    #4-293    #4-294    #4-300
               #5-308    #5-309    #5-310    #5-314    #5-317    #5-324    #5-325    #5-326    #5-327    #5-328
               #5-329    #5-330    #5-331    #5-332    #5-333    #5-334    #5-338    #5-345    #5-346    #5-347
               #5-353    #5-354    #5-355    #5-356    #5-357    #5-358    #5-364    #5-365    #5-368    #5-369
               #5-370    #5-378    #5-381    #5-423    #5-424    #5-425    #5-426    #5-427    #5-428    #5-429
               #5-430    #6-438    #6-439    #6-440    #6-441    #6-442    #6-443    #6-444    #6-445    #6-446
               #6-447    #6-448    #6-449    #6-450    #6-451    #6-453    #6-459    #6-466    #6-469    #6-470
               #6-471    #6-472    #6-473    #6-474    #6-475    #6-476    #6-477    #6-478    #6-479    #6-480
               #6-481    #6-493    #6-502    #6-505    #6-529    #6-539    #6-546    #6-556    #6-565    #6-572
               #6-576    #6-589    #6-615    #6-630    #6-631    #6-632    #6-635    #6-648    #6-654    #6-662
               #6-666    #6-670    #6-674    #6-681    #6-685    #6-705    #6-710    #6-711    #6-721    #6-746
               #6-747    #7-767    #7-778    #7-779    #7-780    #7-781    #7-782    #7-783    #7-784    #7-785
               #7-786    #7-787    #7-788    #7-789    #7-790    #7-800    #7-801    #7-805    #7-811    #7-816
               #7-820    #7-821    #7-822    #7-823    #7-824    #7-825    #7-826    #7-827    #7-828    #7-829
               #7-830    #7-831    #7-832    #7-833    #7-834    #7-835    #7-836    #7-837    #7-838    #7-839
               #7-840    #7-841    #7-842    #7-843    #7-844    #7-850    #7-854    #7-858    #7-865    #7-869
               #7-870    #7-871    #7-872    #7-882    #7-892    #7-898    #7-905    #7-913    #7-916    #7-922
               #7-923    #7-924    #7-931    #7-936    #7-945    #7-959    #7-970    #7-974    #7-991    #7-1016
               #7-1026   #7-1036   #7-1050   #7-1054   #7-1058   #7-1062   #7-1063   #8-1087   #8-1094   #8-1100
               #8-1101   #8-1110   #9-1157   #9-1164   #9-1170   #9-1171   #9-1172   #9-1173   #9-1174   #9-1175
               #9-1176   #9-1177   #9-1178   #9-1179   #9-1180   #9-1181   #9-1182   #9-1183   #9-1184   #9-1185
               #9-1193   #9-1196   #9-1206   #9-1217   #9-1218   #9-1228   #9-1316   #9-1317   #9-1324   #9-1331
               #9-1337   #9-1338   #9-1339   #9-1340   #9-1341   #9-1342   #9-1343   #9-1344   #9-1345   #9-1346
               #9-1347   #9-1348   #9-1349   #9-1350   #9-1359   #9-1360   #9-1385   #9-1394   #9-1407   #9-1419
               #9-1437   #9-1441   #9-1442   #9-1443   #9-1444   #9-1445   #9-1446   #9-1447   #9-1448   #9-1449
               #9-1450   #9-1451   #9-1452   #9-1453   #9-1454   #9-1455   #9-1522   #9-1531   #9-1550   #9-1554
               #9-1558   #9-1562   #9-1571   #10-1602  #10-1608  #10-1609  #10-1610  #10-1611  #10-1612  #10-1613
               #10-1614  #10-1615  #10-1616  #10-1617  #10-1618  #10-1619  #10-1620  #10-1621  #10-1622  #10-1623
               #10-1624  #10-1625  #10-1626  #10-1627  #10-1628  #10-1655  #10-1656  #10-1674  #10-1678  #10-1687
               #10-1688  #10-1697  #10-1698  #10-1712  #10-1716  #10-1782  #10-1795  #10-1803  #10-1808  #10-1817
               #10-1822  #10-1830  #10-1833  #10-1836  #10-1844  #10-1845  #10-1848  #10-1853  #10-1862  #10-1865
               #10-1870  #10-1876  #10-1879  #10-1887  #10-1890  #10-1898  #10-1901  #10-1909  #10-1912  #11-1926
               #11-1932  #11-1933  #11-1934  #11-1951  #11-1960  #11-1969  #11-1970  #12-1998  #12-2004  #12-2005
               #12-2006  #12-2007  #12-2008  #12-2009  #12-2019  #12-2020  #12-2045  #12-2049  #12-2058  #12-2065
               #13-2143  #13-2144  #13-2145  #13-2146  #13-2147  #13-2153  #13-2156  #13-2162  #13-2165  #13-2168
               #13-2176  #13-2179  #13-2185  #13-2188  #13-2191  #13-2197  #13-2202  #13-2205  #13-2206  #13-2207
               #13-2208  #13-2209  #13-2210  #13-2211  #13-2219  #13-2223  #13-2224  #13-2225  #13-2226  #13-2227
               #13-2228  #13-2229  #13-2230  #13-2231  #13-2232  #13-2233  #13-2239  #13-2244  #13-2248  #13-2249
               #13-2250  #13-2251  #13-2258  #13-2263  #13-2267  #13-2268  #13-2397  #13-2423  #13-2424  #14-2437
               #14-2440  #14-2441  #14-2453  #14-2493  #14-2496  #14-2502  #14-2503  #14-2504  #14-2505  #14-2506
               #14-2507  #14-2524  #14-2535  #14-2543  #14-2551  #14-2559  #14-2570  #14-2573  #14-2585  #14-2588
               #14-2621  #14-2634
GCL$           #4-60
ISTAT$         #4-60      4-84
KEYWD$         #4-60      4-88     #4-89     #4-90     #4-91     #4-92     #4-93     #4-94     #4-95     #4-96
               #4-97     #4-98     #4-99     #4-100    #4-101    #4-102    #4-103    #4-104    #4-105    #4-106
               #4-107    #4-108    #4-109    #4-110    #4-111    #4-112    #4-113    #4-114    #4-115    #4-116
               #4-117    #4-118    #4-119    #4-120    #4-121    #4-122    #4-123    #4-124    #4-125    #4-126
               #4-127    #4-128    #4-129    #4-130    #4-131    #4-132    #4-133    #4-134    #4-135    #4-136
               #4-137    #4-138    #4-139    #4-140    #4-141    #4-142    #4-143    #4-144    #4-145    #4-146
               #4-147    #4-148    #4-149    #4-150    #4-151    #4-152    #4-153    #4-154    #4-155    #4-156
               #4-157    #4-158    #4-159    #4-160    #4-161    #4-162    #4-163    #4-164    #4-165    #4-166
```

```
CFEROT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:13     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CLOSE$          #4-93       5-146      5-164
DELET$          #4-93       5-143      5-163
DIR$            #4-93       5-152     #5-165      5-165     #5-166      5-166
EPRINT          #4-79
ERROR$          #4-68       5-158
ERR$            #5-143      5-143     #5-146      5-146     #5-163      5-163     #5-164      5-164
EXIT$S          #4-93       5-166
EXST$S          #4-93       5-165
FDOFF$          #4-96       4-96
FDOF$L          #4-94       4-96
LDFDB$          #5-143      5-143     #5-146      5-146     #5-163      5-163     #5-164      5-164
MOV$            #5-165      5-165
NBOFF$          #4-97       4-97
NBOF$L          #4-94       4-97
OFF$            #4-103     #5-165      5-165      5-165      5-165      5-165
PRINT           #4-63
QDPB$           #4-103      4-103
QIOW$           #4-93       4-103
```

Table of contents

```
     6-  155     $TMSTB - SET UP CETAB TEMPLATE BLOCKS
     7-  204     $TALOC - ALLOCATE TEMPLATES
     8-  223     $T1ALC - ALLOCATE TEMPLATES
```

```
    57                                  ;****
    58                                  ; LIBRARY MACRO CALLS
    59                                  ;****
    60
    61                                          .MCALL  DIR$,CTRDF$,DECTP$
    62
    63 000000                                   CTRDF$                          ; DEFINE NODE COUNTER OFFSETS
    64 000000                                   DECTP$  LIST                    ; DEFINE DEC$DF OFFSETS
                                        ;
                                        ; DEFINE DEC TEMPLATE
                                        ;
       000050                                   .ASECT
               000000                           .=0
       000000                                   .BLKW   1               ; C.LNK
       000002                                   .BLKW   1               ; C.STS
       000004                           C.SEG:  .BLKW   1               ; SEGMENT SIZE
       000006                           C.NDC:  .BLKW   1               ; MAXIMUM NODE COUNTERS
       000010                           C.LKS:  .BLKW   1               ; MAXIMUM LOGICAL LINKS
       000012                           C.INCT: .BLKB   1               ; INCOMING TIMER
       000013                           C.OUTT  .BLKB   1               ; OUTGOING TIMER
       000014                           C.INAC  .BLKB   1               ; INACTIVITY TIMER
       000015                           C.RETF: .BLKB   1               ; RETRANSMIT FACTOR
       000016                           C.DELF: .BLKB   1               ; DELAY FACTOR
       000017                           C.DELW: .BLKB   1               ; DELAY WEIGHT
       000020                           C.IPL:  .BLKB   1               ; INPUT PACKET LIMITER
       000021                           C.LST:  .BLKB   1               ; LINK SERVICE THRESHOLD
               000022                   CD.LEN=.
       000000                                   .PSECT
    65
    66                                  ;****
    67                                  ; DEFINE SLT TEMPLATE
    68                                  ;****
    69 000022                                   .ASECT
    70         000000                           .=0
    71 000000                                   .BLKW   1
    72 000002                                   .BLKW   1
    73 000004                           C.NAM:  .BLKW   1                       ; DDM PROCESS NAME
    74 000006                           C.FLG:  .BLKW   1                       ; FLAGS WORD
    75 000010                           C.CNT:  .BLKB   1                       ; CONTROLLER NUMBER
    76 000011                           C.UNT:  .BLKB   1                       ; UNIT NUMBER
    77 000000                                   .PSECT
    525 001162  103401                  BCS     20$                     ; IF CS, ERROR
    526 001164  000241          17$:    CLC                             ; ELSE, SUCCESS
    527 001166  005226          20$:    INC     (SP)+                   ; CLEAN STACK, PRESERVE CARRY
    528 001170  000207                  RETURN
```

```
**FILE**ID**CFETSB




  CCCCCCCC   FFFFFFFFFF  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  BBBBBBBB
  CCCCCCCC   FFFFFFFFFF  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  BBBBBBBB
CC           FF          EE              TT      SS          BB      BB
CC           FF          EE              TT      SS          BB      BB
CC           FF          EE              TT      SS          BB      BB
CC           FF          EE              TT      SS          BB      BB
CC           FFFFFFFF    EEEEEEEE        TT        SSSSSS    BBBBBBBB
CC           FFFFFFFF    EEEEEEEE        TT        SSSSSS    BBBBBBBB
CC           FF          EE              TT              SS  BB      BB
CC           FF          EE              TT              SS  BB      BB
CC           FF          EE              TT              SS  BB      BB        ....
CC           FF          EE              TT              SS  BB      BB        ....
  CCCCCCCC   FF          EEEEEEEEEE      TT      SSSSSSSS    BBBBBBBB          ....
  CCCCCCCC   FF          EEEEEEEEEE      TT      SSSSSSSS    BBBBBBBB          ....



LL              SSSSSSSS  TTTTTTTTTT
LL              SSSSSSSS  TTTTTTTTTT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL              SSSSSS        TT
LL              SSSSSS        TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LLLLLLLLLL    SSSSSSSS        TT
LLLLLLLLLL    SSSSSSSS        TT
```

```
                                ;
                                ; DEFINE THE PERMANENT PART OF THE TEMPLATES
                                ;****
000022                                  .ASECT
        000000                          .=0
000000                          C.LNK:: .BLKW   1               ; LINK POINTER
000002                          C.STS:: .BLKW   1               ; STATUS WORD
000004                          C.PRM::                         ; PARAMETER LIST (VAR. LENGTH)
002054                                  .PSECT

                                ;
                                ; DEFINE PROCESS TEMPLATE FLAGS WORD BITS (SIMILAR TO PDV FLAG BITS)
                                ;
        000001                          ZF.DDM == 1             ; PROCESS IS A DDM
        000002                          ZF.DLC == 2             ; PROCESS IS A DLC
        000004                          ZF.LLC == 4             ; PROCESS IS AN LLC
        000010                          ZF.MFL == 10            ; PROCESS IS MARKED FOR LOAD
        000020                          ZF.KMX == 20            ; PROCESS IS A KMC DEVICE
        000040                          ZF.MUX == 40            ; PROCESS IS A MUX DEVICE
        000100                          ZF.LMC == 100           ; PROCESS NEEDS MICRO-CODE
        000200                          ZF.TIM == 200           ; PROCESS NEEDS TIMER SUPPORT
        000400                          ZF.MTM == 400           ; PROCESS NEEDS MULTIPLE TIMERS
        001000                          ZF.COU == 1000          ; DLC PROCESS SUPPORTS COUNTERS (DLC$DF)
        002000                          ZF.X3P == 2000          ; LLC PROCESS NEEDS X.25 LEVEL 3 (LLC$DF)
        002000                          ZF.PSE == 2000          ; DDM PROCESS HAS PSEUDO SLT (DDM$DF)
        004000                          ZF.MAN == 4000          ; DLC SUPPORTS NETWORK MANAGEMENT FUNCTIONS (DDM$DF, DLC$DF)
        010000                          ZF.SLI == 10000         ; PROCESS USES THE SYSTEM LEVEL INTERFACE (LLC$DF)
        020000                          ZF.DIA == 20000         ; DEVICE REQUIRES DIAGNOSTIC u-CODE (DDM$DF)
        040000                          ZF.DVP == 40000         ; PROCESS RUNS AT DEVICE PRIORITY (DDM$DF)
        100000                          ZF.INI == 100000        ; PROCESS SUPPORTS INIT/EXIT


                                ;
                                ; DEFINE SLT FLAG BITS
                                ;
        000001                          LF.ENA == 1             ; LINE IS TO BE ENABLED
        000002                          LF.MDC == 2             ; LINE NEEDS MODEM CONTROL
        000004                          LF.MTP == 4             ; LINE IS A MULTI-POINT LINE
        000010                          LF.MFL == 10            ; LINE IS MARKED FOR LOAD
        000020                          LF.TIM == 20            ; LINE NEEDS TIMER SUPPORT
        000040                          LF.X2P == 40            ; LINE NEED X.25 LEVEL 2
        000100                          LF.BRO == 100           ; LINE IS A BROADCAST CHANNEL (ETHERNET)
        000200                          LF.SER == 200           ; LINE HAS SERVICE DISABLED
                                ;
                                ;*****
                                ; DEFINE STATION FLAGS
                                ;
        000100                          SF.ENA == 100           ; STATION IS ENABLED
        000040                          SF.MFL ==  40           ; STATION IS MARKED FOR LOAD (PSI)
        000020                          SF.SVC ==  20           ; STATION IS AN SVC
        000010                          SF.SER ==  10           ; SERVICE DISABLED ON CIRCUIT

                                ;
                                ;*****
                                ; DEFINE THE TEMPLATE STATUS WORD BITS
                                ;       (LOW BYTE CONTAINS TEMPLATE TYPE, HIGH BYTE CONTAINS STATUS FLAGS)
                                ;
```

LC$DF)

```
SYMBOL   VALUE               REFERENCES

OC.CNT  = 000000  G          #11-820       11-825
OC.CST  = 000004  G          #11-794       11-821
OC.CTM  = 000001  G          #11-791
OC.DTE  = 000020  G          #11-797       11-816       11-818
OC.EXE  = 002000  G          #11-815
OC.HTM  = 002000  G          #11-805       11-815       11-819
OC.MBL  = 000100  G          #11-799       11-818
OC.MRC  = 001000  G          #11-803       11-817
OC.MWI  = 000010  G          #11-795       11-818
OC.MXR  = 004000  G          #11-806       11-819
OC.NUM  = 100000  G          #11-810       11-817
OC.NWC  = 000240  G          #11-823
OC.PVC  = 000060  G          #11-816       11-818
OC.RPR  = 040000  G          #11-809       11-819
OC.RTM  = 000400  G          #11-802       11-817
OC.SLT  = 046002  G          #11-819
OC.STA  = 000002  G          #11-792       11-819
OC.STN  = 010204  G          #11-821       11-822       11-826
OC.SVC  = 101400  G          #11-817       11-818
OC.TRI  = 000200  G          #11-800       11-821       11-823
OC.UNT  = 010204  G          #11-822       11-827
OC.USG  = 020000  G          #11-808
OC.X25  = 101570  G          #11-818
OE.ACO  = 040000  G          #11-855       11-871
OE.ADD  = 000002  G          #11-840       11-870
OE.AHO  = 020000  G          #11-854       11-871
OE.AMA  = 010000  G          #11-853       11-871
OE.HOS  = 001000  G          #11-848       11-870
OE.ID   = 000004  G          #11-841       11-870
OE.LNK  = 000010  G          #11-842       11-868
OE.LV2  = 070000  G          #11-871       11-872
OE.MXC  = 000020  G          #11-843       11-872
OE.MXH  = 000040  G          #11-844       11-872
OE.NAM  = 000001  G          #11-839       11-870
OE.NCO  = 000200  G          #11-846       11-868
OE.NN   = 000100  G          #11-845       11-872
OE.NOD  = 001007  G          #11-870
OE.NSP  = 000610  G          #11-868
OE.ROU  = 072160  G          #11-872
OE.RTM  = 002000  G          #11-849       11-872
OE.SUB  = 004000  G          #11-851
OE.VFY  = 000400  G          #11-847       11-868
OL.BAU  = 000040  G          #11-771       11-785
OL.BSZ  = 020000  G          #11-779       11-782
OL.CNT  = 000624  G          #11-783       11-825
OL.CSR  = 000200  G          #11-773       11-783       11-786
OL.CTM  = 000001  G          #11-765       11-781
OL.DPR  = 001000  G          #11-775       11-785
OL.HBT  = 002000  G          #11-776       11-782
OL.NWC  = 000020  G          #11-786
OL.PLX  = 000010  G          #11-769       11-785
OL.PRI  = 000004  G          #11-768       11-783
```

N 2

Table of contents

5- 88 $CFINI - INITIALIZE END OF TASK BUFFERS

```
CFELIS     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE            REFERENCES

BLANK      0001 0 R        4-88       #4-92
DSPTBL     000000 R       #4-68        5-112
GTIM       000114 R       #4-89        5-108
G.TIBA   = 000002          4-89
IO.ATT   = ******   GX     4-87        5-116
IO.DET   = ******   GX     5-114
IO.WVB   = ******   GX     4-88
LFDPB      000064 R       #4-88        5-113
Q.IOFN   = 000002          4-87        4-88       *5-114      *5-116
TIDPB      000034 R       #4-87        5-109      *5-114       5-115      *5-116
$CFLIS     000152 RG      #5-107
$COMP    = ******   GX     5-110
$GTBUF     000130 RG       4-89       #4-91
$HDRBK     000120 RG      #4-90
$SHCIR   = ******   GX     4-76        4-77
$SHEXE   = ******   GX     4-71
$SHLIN   = ******   GX     4-74
$SHLOG   = ******   GX     4-75
$SHOBJ   = ******   GX     4-73
$SHPRO   = ******   GX     4-70
$SHREM   = ******   GX     4-69
$SHSNA   = ******   GX     4-82
$SHSYS   = ******   GX     4-72
$SHXAC   = ******   GX     4-80
$SHXPR   = ******   GX     4-78
$SHXSV   = ******   GX     4-79
$SHX29   = ******   GX     4-81
$$$ARG   = 000003         #4-87        4-87        4-87       #4-88        4-88       #4-88        4-88       #4-88        4-88
                          #4-88        4-88        4-88

$$$GLB   = ******          4-87        4-88        4-89
```

Table of contents

```
     4-   53     MACRO DEFINITIONS
     5-   83     LOCAL DATA
     6-  129     LOCAL SYMBOL DEFINITIONS
     7-  149     $CFPRG - PERFORM PURGE COMMAND
     8-  166     PUROBJ - PURGE OBJECT
     9-  215     PURLOG - PURGE LOGGING
    10-  301     PURLIN  - PURGE LINE
    11-  319     PURCIR  - PURGE CIRCUIT
    12-  402     QREM - REMOVE ENTRY FROM QUEUE
```

```
CFEPRG - PERFORM PURGE COMMAND  MACRO V05.03b Saturday 29-Jun-85 04:57  Page 12-1
Symbol table

CO.KNC= ****** GX     FF.CIR= 000040        F.FLG   000014        P$VCT = ****** GX     $COMP = ****** GX
CS.EVT= ****** GX     FF.CON= 000001        F.LEN   000016        P$VLEN= ****** GX     $EVENT= ****** GX
CS.OBJ= ****** GX     FF.FIL= 000002        F.LENX= ****** GX     P$VNAM= ****** GX     $FLAGS= ****** GX
CS.PVC= ****** GX     FF.HST= 040000        F.LIN   000020        P$VOWN= ****** GX     $KNEVT= ****** GX
CS.SVC= ****** GX     FF.LIN= 000010        F.LNK   000000        QREM    001244R       $OPRND= ****** GX
C$LENX= ****** GX     FF.MOD= 000100        F.MOD   000022        SNKTBL  000202R       $PUREM= ****** GX
C.CLS   000004        FF.MON= 000004        F.REM   000026        S$LCTT= ****** GX     $PUSNA= ****** GX
C.EVT   000006        FF.MSK= 000077        F.SEV   000004        TEMP    000212R       $PUXAC= ****** GX
C.PRM = ****** GX     FF.PRT= 000200        OBJLEN= 000013        WC.EVE= ****** GX     $PUXPR= ****** GX
C.SNK   000016        FF.OUL= 000370        OC.CTM= ****** GX     WC.LOG= ****** GX     $PUXSV= ****** GX
C.STS = ****** GX     FF.REM= 100000        OM.CON= ****** GX     WC.OBJ= ****** GX     $PUXS9= ****** GX
DSPTBL  000146R       FL.DLM= ****** GX     OM.FIL= ****** GX     X$LENX= ****** GX     $RLBF = ****** GX
ERRCTY  000044R       FL.X25= ****** GX     OM.KEV= ****** GX     $BYTMN= ****** GX     $TMLST= ****** GX
ERRNSC  000101R       F.ADD   000016        O.LENX= ****** GX     $BYTXT= ****** GX     $TNUM = ****** GX
ERROBJ  000000R       F.CEV   000030        PURCIR  000756R       $CFERR= ****** GX     $WILD = ****** GX
EVTLEN  000020        F.CIR   000024        PURLIN  000734R       $CFPRG  000214RG      $XXAVL= ****** GX
EX$ERR= 000002        F.CLS   000002        PURLOG  000360R       $CIRID= ****** GX     .FNSLT= ****** GX
FF.ADD= 000020        F.EVT   000004        PUROBJ  000226R       $CLASS= ****** GX

. ABS.  000040    000   (RW,I,GBL,ABS,OVR)
        001256    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9209 Words  ( 36 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:10.38
SY:CFEPRG.V2,[132,134]CFEPRG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEPRG
```

```
    359                                ;
    360                                ; LIST OPTIONS FOR X25-PROTOCOL MODULE
    361                                ;
    362 000000                                 STATE$  LIXPR
    363 000000                                  TRAN$  !LISOPT,$EXIT
    364 000000                                  TRAN$  !LXPRAB,LISOPT
    365 000000                                  TRAN$  $LAMDA
    366 000000                                 STATE$  LXPRAB
    367 000000                                  TRAN$  !KNG,    $EXIT,,          XP.KNG, $OPRND
    368 000000                                  TRAN$  !CUG,    $EXIT,,          XP.CUG, $OPRND
    369 000000                                  TRAN$  !KND,    $EXIT,,          XP.KND, $OPRND
    370 000000                                  TRAN$  !DTE,    $EXIT,,          XP.DTE, $OPRND
    371                                ;
    372                                ; LIST OPTIONS FOR X25/X29-SERVER, X25-ACCESS MODULES
    373                                ;
    374 000000                                 STATE$  LIXAC   ;;;,     &P$LISO,ALLOWNULL
    375 000000                                  TRAN$  $LAMDA
    376 000000                                 STATE$  LIXSR
    377 000000                                  TRAN$  !LISOPT,$EXIT
    378 000000                                  TRAN$  !LXSRAB,LISOPT
    379 000000                                 STATE$  LXSRAB
    380 000000                                  TRAN$  !DST,    $EXIT,,          XS.DST, $OPRND
    381 000000                                  TRAN$  !KNDS,   $EXIT,,          XS.KND, $OPRND
    382                                ;
    383                                ; PARSE KNOWN GROUPS
    384                                ;
    385 000000                                 STATE$  KNG
    386 000000                                  TRAN$  &K$KNO
    387 000000                                 STATE$  ,&P$PEN
    388 000000                                  TRAN$  &K$GRPS,$EXIT
    389                                ;
    390                                ; PARSE CLOSED USER GROUP
    391                                ;
    392 000000                                 STATE$  CUG
    393 000000                                  TRAN$  &K$GRP
    394 000000                                 STATE$  ,&P$CUG
    395 000000                                  TRAN$  $STRNG, $EXIT,  .MOCUG
    396                                ;
    397                                ; PARSE KNOWN DTES
    398                                ;
    399 000000                                 STATE$  KND
    400 000000                                  TRAN$  &K$KNO
    401 000000                                 STATE$  ,&P$PEN
    402 000000                                  TRAN$  &K$DTES,$EXIT
    403
    404                                ;
    405                                ; PARSE KNOWN DESTINATIONS
    406                                ;
    407 000000                                 STATE$  KNDS
    408 000000                                  TRAN$  &K$KNO
    409 000000                                 STATE$  ,&P$PEN
    410 000000                                  TRAN$  &K$DSTS,$EXIT
    411                                ;
    412                                ; PARSE DESTINATION
    413                                ;
    414 000000                                 STATE$  DST
    415 000000                                  TRAN$  &K$DST
```

```
000000                              STATE$  NODADD
000000                               TRAN$  $LAMDA,,          $BL.MAN
000000                              STATE$
000000                               TRAN$  $LAMDA,,          .RNINI

000000                              STATE$
000000                               TRAN$  $DNUMB,NODN1,     .RNADD
000000                               TRAN$  $LAMDA, NODFL

000000                              STATE$  NODN1
000000                               TRAN$  <'.>,    PH4AD
000000                               TRAN$  $LAMDA, NODND

000000                              STATE$  PH4AD
000000                               TRAN$  $DNUMB,NODND,     .RNAD2
000000                               TRAN$  $LAMDA, NODFL

000000                              STATE$  NODND
000000                               TRAN$  $EOS,    NODN2
000000                               TRAN$  $BLANK, NODN2
000000                               TRAN$  $LAMDA

000000                              STATE$  NODFL
000000                               TRAN$  $LAMDA, $EXIT,  .BL.AU3

000000                              STATE$  NODN2
000000                               TRAN$  $LAMDA, $EXIT,  $BL.AUT


                          ;
                          ; KNOWN NODES
                          ;
000000                              STATE$  KNNOD
000000                               TRAN$  &K$KNO

000000                              STATE$  ,&P$PEN
000000                               TRAN$  &K$NODS,$EXIT,,WC.NOD,$WILD
```

```
                        ;
                        ; LOOK FOR VERIFICATION
                        ;
000000                          STATE$  EVFY
000000                           TRAN$  &K$VER

000000                          STATE$
000000                           TRAN$  &K$STA, VFY1
000000                           TRAN$  $LAMDA

000000                          STATE$  VFY1,&P$STA
000000                           TRAN$  &K$ON,   $EXIT,, SW.VFY, $SWICH
000000                           TRAN$  &K$OFF,  $EXIT
                        ;
                        ; LOOK FOR HOP MAXIMUM
                        ;
000000                          STATE$  EHOP
000000                           TRAN$  &K$MAX

000000                          STATE$  ,&P$MAX
000000                           TRAN$  &K$HOP

000000                          STATE$  ,&P$HOP
000000                           TRAN$  $DNUMB, $EXIT,  .EXHOP
                        ;
                        ; LOOK FOR COST MAXIMUM
                        ;
000000                          STATE$  ECOST
000000                           TRAN$  &K$MAX

000000                          STATE$  ,&P$MAX
000000                           TRAN$  &K$COS

000000                          STATE$  ,&P$COS
000000                           TRAN$  $DNUMB, $EXIT,  .EXCOS
                        ;
                        ; LOOK FOR ADDRESS MAXIMUM
                        ;
000000                          STATE$  ENN
000000                           TRAN$  &K$MAX

000000                          STATE$  ,&P$MAX
000000                           TRAN$  &K$ADD

000000                          STATE$  ,&P$ADD
000000                           TRAN$  $DNUMB, $EXIT,  .EXNN
                        ;
                        ; LOOK FOR MAXIMUM NODE COUNTERS
                        ;
000000                          STATE$  ENCOU
000000                           TRAN$  &K$MAX

000000                          STATE$  ,&P$MAX
000000                           TRAN$  &K$NOD

000000                          STATE$  ,&P$NOD
000000                           TRAN$  &K$COU
```

```
     000000                           STATE$  NNAM
     000000                             TRAN$  &K$NET
     000000                           STATE$  ,&P$NNO
     000000                             TRAN$  &K$NAM
     000000                           STATE$  ,&P$NNA
     000000                             TRAN$  $STRNG,$EXIT,.MONNA

                              ;
                              ; SUBEXPRESSION TO PARSE MAXIMUM CIRCUITS (MODULE CHARACTERISTICS)
                              ;
     000000                           STATE$  MCIR
     000000                             TRAN$  &K$MAX
     000000                           STATE$  ,        &P$PEN
     000000                             TRAN$  &K$CIRS
     000000                           STATE$  ,        &P$MCIR
     000000                             TRAN$  $DNUMB, $EXIT,  .MOMC1
                              ;
                              ; SUBEXPRESSION TO PARSE MAXIMUM CIRCUITS (DTE)
                              ;
     000000                           STATE$  MCIRD
     000000                             TRAN$  &K$MAX
     000000                           STATE$  ,        &P$PEN
     000000                             TRAN$  &K$CIRS
     000000                           STATE$  ,        &P$MCIR
     000000                             TRAN$  $DNUMB, $EXIT,  .MOMCI
                              ;
                              ; SUBEXPRESSION TO PARSE DTE STATE
                              ;
     000000                           STATE$  DSTA
     000000                             TRAN$  &K$STA
     000000                           STATE$  ,        &P$STA
     000000                             TRAN$  &K$OFF, $EXIT
     000000                             TRAN$  &K$ON,  $EXIT,, XP.ST1, $SWICH
                              ;
                              ; SUBEXPRESSION TO PARSE GROUP NUMBER
                              ;
     000000                           STATE$  GNUM
     000000                             TRAN$  &K$NUM
     000000                           STATE$  ,        &P$NUM
     000000                             TRAN$  $DNUMB, $EXIT,  .MONUM
                              ;
                              ; SUBEXPRESSION TO PARSE TYPE BILATERAL
                              ;
     000000                           STATE$  TBIL
     000000                             TRAN$  &K$TYP
     000000                           STATE$  ,        &P$TYP
     000000                             TRAN$  &K$BIL, $EXIT
                              ;
                              ; SUBEXPRESSION TO PARSE CHANNEL LIST
                              ;
     000000                           STATE$  CHNL
     000000                             TRAN$  &K$CHNS,,        .MOINC
     000000                           STATE$  ,        &P$CHNL
     000000                             TRAN$  $LAMDA
     000000                           STATE$  CHNL1
     000000                             TRAN$  $DNUMB,,         .MOCNN
     000000                           STATE$
```

```
                                                              M 11
CFEFRS     CREATED BY  MACRO  ON 2-JUL-85 AT 14:24       PAGE 3
MACRO CROSS REFERENCE                                    CREF   04.00
MACRO NAME     REFERENCES

                  #4-167    #4-168    #4-169    #4-170    #4-171    #4-172    #4-173    #4-174    #4-175    #4-176
                  #4-177    #4-178    #4-179    #4-180    #4-181    #4-182    #4-183    #4-184    #4-185    #4-186
                  #4-187    #4-188    #4-189    #4-190    #4-191    #4-192    #4-193    #4-194    #4-195    #4-196
                  #4-197    #4-198    #4-199    #4-200    #4-201    #4-202    #4-203    #4-204    #4-205    #4-206
                  #4-207    #4-208    #4-209    #4-210    #4-211    #4-212    #4-213    #4-214    #4-215    #4-216
                  #4-217    #4-218    #4-219    #4-220    #4-221    #4-222    #4-223    #4-224    #4-225    #4-226
                  #4-227    #4-228    #4-229    #4-230    #4-231    #4-232    #4-233    #4-234    #4-235    #4-236
                  #4-237    #4-238    #4-239    #4-240    #4-241    #4-242    #4-243    #4-244    #4-245    #4-246
                  #4-247    #4-248    #4-249    #4-250    #4-254    #4-255    #4-256    #4-257    #4-258    #4-259
                  #4-260    #4-261    #4-262    #4-263    #4-264    #4-265    #4-266    #4-267    #4-268    #4-269
                  #4-270    #4-271    #4-272    #4-273    #4-274    #4-275    #4-276    #4-277    #4-278    #4-279
MTRAN$            #4-84
STATE$            #4-60      4-285    #4-298     5-306    #5-312    #5-315    #5-319    #5-322    #5-336    #5-343
                  #5-351    #5-362    #5-366    #5-374    #5-376    #5-379    #5-385    #5-387    #5-392    #5-394
                  #5-399    #5-401    #5-407    #5-409    #5-414    #5-416    #5-421     6-436    #6-457    #6-463
                  #6-467    #6-485    #6-491    #6-494    #6-500    #6-503    #6-506    #6-512    #6-514    #6-520
                  #6-522    #6-527    #6-530    #6-535    #6-537    #6-544    #6-547    #6-552    #6-554    #6-561
                  #6-563    #6-570    #6-574    #6-578    #6-584    #6-587    #6-594    #6-597    #6-603    #6-608
                  #6-613    #6-619    #6-621    #6-626    #6-628    #6-633    #6-639    #6-644    #6-646    #6-652
                   6-660    #6-664    #6-668    #6-672    #6-676    #6-679    #6-683    #6-687    #6-690     6-697
                  #6-699    #6-701    #6-703    #6-706    #6-708    #6-712    #6-714    #6-719    #6-725    #6-727
                  #6-729    #6-734     6-741    #6-744    #6-751    #6-754     7-762    #7-765    #7-769    #7-774
                  #7-776    #7-794    #7-796    #7-798    #7-803    #7-809    #7-812    #7-814    #7-818    #7-848
                  #7-852    #7-856    #7-860    #7-863    #7-866    #7-876    #7-878    #7-880    #7-886    #7-888
                  #7-890    #7-896    #7-899    #7-901    #7-903    #7-909    #7-911    #7-914    #7-918    #7-920
                  #7-929    #7-932    #7-934    #7-940    #7-943    #7-947    #7-949    #7-954    #7-957    #7-961
                  #7-963    #7-968    #7-972    #7-976    #7-979    #7-981    #7-986    #7-989    #7-993    #7-995
                  #7-1000   #7-1003   #7-1006   #7-1008   #7-1014   #7-1021   #7-1024   #7-1028   #7-1031   #7-1034
                  #7-1038   #7-1043   #7-1045   #7-1048   #7-1052   #7-1056   #7-1060   #7-1065   #7-1068   #7-1075
                  #7-1078    8-1085   #3-1089   #8-1092   #8-1098   #8-1105   #8-1108   #8-1114   #8-1117   #8-1120
                  #8-1125   #8-1128   #8-1131    8-1137   #8-1140   #8-1145   #8-1148    9-1155   #9-1159   #9-1162
                  #9-1168   #9-1189   #9-1191   #9-1194   #9-1197   #9-1202   #9-1204   #9-1207   #9-1213   #9-1215
                  #9-1223   #9-1226   #9-1232   #9-1235   #9-1240   #9-1243   #9-1249   #9-1252   #9-1255   #9-1260
                  #9-1262   #9-1267   #9-1269   #9-1271   #9-1276   #9-1278   #9-1280   #9-1285   #9-1287   #9-1289
                  #9-1294   #9-1296   #9-1298   #9-1303   #9-1305   #9-1307   #9-1312   #9-1314    9-1322   #9-1326
                  #9-1329   #9-1335   #9-1354   #9-1357   #9-1364   #9-1367   #9-1372   #9-1375   #9-1380   #9-1383
                  #9-1389   #9-1392   #9-1398   #9-1400   #9-1405   #9-1409   #9-1412   #9-1417   #9-1421   #9-1424
                  #9-1429   #9-1432   #9-1435   #9-1439   #9-1459   #9-1462   #9-1465   #9-1470   #9-1472   #9-1474
                  #9-1479   #9-1481   #9-1483   #9-1488   #9-1490   #9-1492   #9-1497   #9-1499   #9-1501   #9-1506
                  #9-1508   #9-1510    9-1517   #9-1520   #9-1526   #9-1529   #9-1536   #9-1542   #9-1545   #9-1548
                  #9-1552   #9-1556   #9-1560   #9-1564   #9-1566   #9-1569   #9-1576   #9-1579   #9-1584   #9-1587
                   10-1594  #10-1597  #10-1600  #10-1606  #10-1632  #10-1635  #10-1640  #10-1643  #10-1648  #10-1651
                  #10-1653  #10-1661  #10-1664  #10-1667  #10-1672  #10-1676  #10-1682  #10-1685  #10-1692  #10-1695
                  #10-1700  #10-1702  #10-1707  #10-1710  #10-1714  #10-1720  #10-1723  #10-1726  #10-1731  #10-1734
                  #10-1737  #10-1742  #10-1745  #10-1748  #10-1753  #10-1756  #10-1759  #10-1762  #10-1767  #10-1769
                  #10-1771  #10-1776  #10-1778  #10-1780  #10-1783  #10-1789  #10-1791  #10-1793  #10-1799  #10-1801
                  #10-1804  #10-1806  #10-1813  #10-1815  #10-1818  #10-1820  #10-1826  #10-1828  #10-1831  #10-1834
                  #10-1842  #10-1846  #10-1849  #10-1851  #10-1858  #10-1860  #10-1863  #10-1866  #10-1872  #10-1874
                  #10-1877  #10-1883  #10-1835  #10-1888  #10-1894  #10-1896  #10-1899  #10-1905  #10-1907  #10-1910
                   11-1918  #11-1921  #11-1924  #11-1930  #11-1938  #11-1941  #11-1946  #11-1949  #11-1955  #11-1958
                  #11-1964  #11-1967  #11-1974  #11-1977  #11-1980  #11-1983   12-1990  #12-1993  #12-1996  #12-2002
                  #12-2014  #12-2017  #12-2024  #12-2027  #12-2032  #12-2035  #12-2040  #12-2043  #12-2047  #12-2051
```

N 11

```
**FILE**ID**CFERQB
```

```
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   RRRRRRRR         QQQQQQ     BBBBBBBB
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   RRRRRRRR         QQQQQQ     BBBBBBBB
CC           FF           EE           RR      RR   QQ      QQ   BB      BB
CC           FF           EE           RR      RR   QQ      QQ   BB      BB
CC           FF           EE           RR      RR   QQ      QQ   BB      BB
CC           FF           EE           RR      RR   QQ      QQ   BB      BB
CC           FFFFFFFF     EEEEEEEE     RRRRRRRR     QQ      QQ   BBBBBBBB
CC           FFFFFFFF     EEEEEEEE     RRRRRRRR     QQ      QQ   BBBBBBBB
CC           FF           EE           RR  RR       QQ  QQ  QQ   BB      BB
CC           FF           EE           RR  RR       QQ  QQ  QQ   BB      BB
CC           FF           EE           RR    RR     QQ    QQ     BB      BB       ....
CC           FF           EE           RR    RR     QQ    QQ     BB      BB       ....
  CCCCCCCC   FF           EEEEEEEEEE   RR      RR     QQQQ  QQ   BBBBBBBB         ....
  CCCCCCCC   FF           EEEEEEEEEE   RR      RR     QQQQ  QQ   BBBBBBBB         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
			.TITLE	CFESTB - STATE TABLES TO READ IN CETAB FILE
			.IDENT	/V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;	STATE TABLE AND ACTION ROUTINES TO DETERMINE WHICH DEFINITION IT IS.
;
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	10-FEB-78
;	VERSION 2.0 RELEASE
;
; 2.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;
```

```
                                        ;
                                        ; REDEFINE CONSTANTS FOR BYTE AREA ALLOCATION
                                        ; ** NTL ALLOCATES ALL BLOCKS  ON A 4 BYTE BOUNDARY
                                        ;
                                                .MACRO  REDEF   S, M, R
                                                .MCALL  'M
                                                'M
                                                .IF NB <'R>
                                        'R == < <'S+3> & <^C3> >
                                                .IFF
                                        'S'X == < <'S+3> & <^C3> >
                                                .ENDC
                                                .ENDM

 000000                                         REDEF   O.LEN, OBJDF$           ; LENGTH OF OBJECT BLOCK
 000000                                         REDEF   R.LEN, RNBDF$           ; LENGTH OF REMOTE NODE BLOCK
 000000                                         REDEF   F.LEN, FLTDF$           ; LENGTH OF FILTER BLOCK
 000000                                         REDEF   L$LEN, DTEDF$           ; LENGTH OF DTE DESCRIPTOR BLOCK
 000000                                         REDEF   C$LEN, PVCDF$           ; LENGTH OF PVC NAME BLOCK
 000000                                         REDEF   G$LEN, CUGDF$           ; LENGTH OF CUG BLOCK
 000000                                         REDEF   R$LEN, RDTDF$           ; LENGTH OF REMOTE DTE BLOCK
 000000                                         REDEF   X$LEN, XCBDF$           ; LENGTH OF X25 CIRCUIT BLOCK
 000000                                         REDEF   D$FLEN,DSTDF$,D$FLNX    ; FIXED PART OF DESTINATION BLOCK
 000000                                         REDEF   S$LEN, SVCDF$           ; LENGTH OF SVC DESCRIPTOR
 000000                                         REDEF   S.LENG, SERDF$,S.LENX   ; LENGTH OF SER$DF DESCRIPTOR
                                        ;
                                        ; THE LLT LENGTH WILL **ALWAYS** INCLUDE THE SLI EXTENSION, BECAUSE
                                        ; CFE IS DISTRIBUTED AS A PREBUILT TASK.  THE EXTRA SPACE WILL PROBABLY
                                        ; BE USED FOR ALIASES, ADDITIONAL REMOTE NODES, OBJECTS AND X.25 STRUCTURES
                                        ; AFTER THE SYSTEM COMES UP.
                                        ;
                000001                          N$$SLI  = 1                     ; FORCE SLI EXTENSION
 000000                                         REDEF   L.LNG, LLTDF$           ; LENGTH OF AN LLT
```

```
                                        .SBTTL  $FNSTA - FIND AN STATION TEMPLATE
                                ;+
                                ; *** - $FNSTA - FIND AN STATION TEMPLATE
                                ;
                                ; INPUTS:
                                ;       R1 = ADDRESS OF CONTEXT BLOCK
                                ;       R5 = POINTER TO CURRENT TEMPLATE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       R5 = ADDRESS OF TEMPLATE
                                ;
                                ;-

001172                          $FNSTA::
001172  011505                  10$:    MOV     (R5),R5                 ; GET NEXT TEMPLATE
001174  000261                          SEC                             ; ASSUME FAILURE
001176  001421                          BEQ     20$                     ; IF EQ, END OF LIST
001200  122765  000000G 000000G         CMPB    #CS.STA,C.STS(R5)       ; IS THIS A STATION TEMPLATE ?
001206  000261                          SEC                             ; ASSUME NOT
001210  001014                          BNE     20$                     ; IF NE, NO .. NOT HERE
001212  032767  000000G 000000G         BIT     #WC.TRI,$WILD           ; WILD TRIBUTARIES ?
001220  001004                          BNE     15$                     ; IF NE, YES
001222  126165  000000G 000000G         CMPB    L$TRI(R1),C.PRM(R5)     ; IS THIS THE RIGHT TRIBUTARY ?
001230  001360                          BNE     10$                     ; IF NE, NO .. KEEP LOOKING
001232  116567  000000G 000000G 15$:    MOVB    C.PRM(R5),$STAN         ; SAVE THE CURRENT STATION NUMBER
001240  000241                          CLC                             ; INDICATE SUCCESS
001242  000207                  20$:    RETURN
```

Table of contents

```
     4-    57     DATA AREA
     5-    69     $GTNUM - GET NUMBER INTO TEMPLATE (WORD)
     5-    70     $GTNMB - GET NUMBER INTO TEMPLATE (BYTE)
     5-   113     $GTR50 - GET RAD50
     5-   169     $CLERR - CLEAR PARSE ERROR
```

```
	000001			CS.BUF == 1		; BUF$DF TEMPLATE
	000002			CS.SLT == 2		; SLT$DF TEMPLATE
	000003			CS.LLC == 3		; LLC$DF TEMPLATE
	000004			CS.DLC == 4		; DLC$DF TEMPLATE
	000005			CS.DDM == 5		; DDM$DF TEMPLATE
	000006			CS.CNT == 6		; CNT$DF TEMPLATE
	000007			CS.UNT == 7		; UNT$DF TEMPLATE
	000010			CS.STA == 10		; STA$DF TEMPLATE
	000011			CS.PAR == 11		; PAR$DF TEMPLATE
	000012			CS.NOD == 12		; NOD$DF TEMPLATE
	000013			CS.REM == 13		; REM$DF TEMPLATE
	000014			CS.OBJ == 14		; OBJ$DF TEMPLATE
	000015			CS.FEA == 15		; FEA$DF TEMPLATE
	000016			CS.ROU == 16		; ROU$DF TEMPLATE
	000017			CS.EVT == 17		; EVT$DF TEMPLATE
	000020			CS.LOG == 20		; LOG$ST TEMPLATE
				;
				; PSI TEMPLATES  ** ORDER MUST NOT BE CHANGED **
				;
	000021			CS.SVC == 21		; SVC$DF TEMPLATE
	000022			CS.PSN == 22		; PSN$DF TEMPLATE
	000023			CS.X3P == 23		; X3P$DF TEMPLATE
	000024			CS.X2P == 24		; X2P$DF TEMPLATE
	000025			CS.X29 == 25		; X29$DF TEMPLATE
	000026			CS.DTE == 26		; DTE$DF TEMPLATE
	000027			CS.CHN == 27		; CHN$DF TEMPLATE
	000030			CS.PVC == 30		; PVC$DF TEMPLATE
	000031			CS.CUG == 31		; CUG$DF TEMPLATE
	000032			CS.DST == 32		; DST$DF TEMPLATE
	000033			CS.DSA == 33		; DSA$DF TEMPLATE
	000034			CS.DSC == 34		; DSC$DF TEMPLATE
	000035			CS.RDT == 35		; RDT$DF TEMPLATE
							; 36 (RESERVED FOR POSSIBLE
							; 37  FUTURE USE)

				;
				; SNA GATEWAY TEMPLATES (ORDER NOT IMPORTANT)
				;
	000040			CS.SNA == 40		; SNA$DF TEMPLATE
	000041			CS.SAC == 41		; SAC$DF TEMPLATE
	000042			CS.SAD == 42		; SAD$DF TEMPLATE
	000043			CS.SPU == 43		; SPU$DF TEMPLATE

				;
				; EXTENDED REMOTE NODE DEFINITION TEMPLATES
				;
	000044			CS.SER  == 44		; SER$DF TEMPLATE
	000045			CS.FIL  == 45		; FIL$DF TEMPLATE
				;
				; DEC$DF TEMPLATE
				;
	000046			CS.DEC  == 46		; DEC$DF TEMPLATE

				;
				; COMMON STATUS FLAGS
				;
	100000			CS.MOD  == 100000	; =1 IF TEMPLATE WAS [illegible]IFIED
```

```
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE               REFERENCES

OL.RTC   = 004000   G        #11-777      11-782
OL.RTT   = 040000   G        #11-780      11-782
OL.SEC   = 000100   G        #11-772      11-785      11-786
OL.SLT   = 000003   G        #11-781
OL.STA   = 000002   G        #11-766      11-781
OL.STN   = 000000   G        #11-784      11-785      11-826
OL.UNT   = 001150   G        #11-785      11-827
OL.URM   = 000400   G        #11-774      11-783
OL.VEC   = 000020   G        #11-770      11-783      11-786
OL.WSZ   = 010000   G        #11-778      11-782
OL.X25   = 076000   G        #11-782
OM.CON   = 000004   G        #11-930
OM.EVE   = 000020   G        #11-932
OM.FIL   = 000010   G        #11-931
OM.KEV   = 000040   G        #11-933
OM.MON   = 000002   G        #11-929
OM.STA   = 000001   G        #11-928
ON.ADD   = 000001   G        #11-893
ON.NAM   = 000001   G        #11-892
OO.COP   = 000400   G        #11-922
OO.NAM   = 002000   G        #11-924
OO.USR   = 000200   G        #11-921
OO.VFY   = 001000   G        #11-923
OP.NCT   = 000004   G        #11-835
OP.NLT   = 000002   G        #11-834
OP.STA   = 000001   G        #11-833
OS.BLK   = 001000   G        #11-886      11-887
OS.BUF   = 000370   G        #11-888
OS.BYT   = 000400   G        #11-885      11-887
OS.CCB   = 000010   G        #11-880      11-888
OS.LOC   = 000001   G        #11-877      11-887
OS.NAM   = 000002   G        #11-878      11-887
OS.PAR   = 000004   G        #11-879      11-887
OS.PRT   = 001407   G        #11-887
OS.RBN   = 000020   G        #11-881      11-888
OS.RBS   = 000040   G        #11-882      11-888
OS.SBN   = 000100   G        #11-883      11-888
OS.THR   = 000200   G        #11-884      11-888
PV$OWN   = 000400   G        #12-1165
PV$PSZ   = 001000   G        #12-1166
PV$WSZ   = 002000   G        #12-1167
P$SFLG     000012   G        #9-465
P$SLEN   = 000020   G        #9-468
P$SNAM     000004   G        #9-463
P$SPOR     000010   G        #9-464
P$STHI     000016   G        #9-467
P$STLO     000014   G        #9-466
P$VCT      000012   G        #9-499
P$VFLG     000022   G        #9-503
P$VLCN     000010   G        #9-498
P$VLEN   = 000024   G        #9-505
P$VNAM     000004   G        #9-497
```

```
                .TITLE  CFEINI - ONCE ONLY INITIALIZATION
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - END OF TASK INITIALIZATION
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; AUTHOR: ROD GAMACHE
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  14-DEC-79
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;

```

```
CFELIS     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME       REFERENCES

DIR$             #4-60       5-108       5-109       5-113       5-115
GTIM$            #4-60        4-89
OFF$             #4-87      #4-88       #4-89
QDPB$            #4-87        4-87      #4-88         4-88
QIOW$            #4-60        4-87        4-88
```

```
                                        .TITLE  CFEPRG - PERFORM PURGE COMMAND
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; CFE - PERFORM PURGE COMMAND
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  14-DEC-79
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;

```

```
CFEPRG     CREATED BY  MACRO  ON 29-JUN-85 AT 04:57     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE              REFERENCES

CO.KNC   = ******  GX        11-335       11-353
CS.EVT   = ******  GX        9-263
CS.OBJ   = ******  GX        8-187
CS.PVC   = ******  GX        11-351
CS.SVC   = ******  GX        11-382
C$LENX   = ******  GX        11-376       11-377
C.CLS      000004           #6-142        9-265
C.EVT      000006           #6-143        9-270
C.PRM    = ******  GX        8-191
C.SNK      000016           #6-144        9-267
C.STS    = ******  GX        8-187        9-263        11-351       11-382
DSPTBL     000146 R         #5-101        7-164
ERRCTY     000044 R         #5-93         11-397
ERRNSC     000101 R         #5-94         11-399
ERROBJ     000000 R         #5-92         8-212
EVTLEN     000020           #6-145        9-290
EX$ERR   = 000002           #5-91         5-92         5-93         5-94
FF.ADD   = 000020           #4-81         4-81
FF.CIR   = 000040           #4-81         4-81
FF.CON   = 000001           #4-81         5-121        9-246
FF.FIL   = 000002           #4-81         5-122        9-249
FF.HST   = 040000           #4-81
FF.LIN   = 000010           #4-81         4-81
FF.MOD   = 000100           #4-81         4-81
FF.MON   = 000004           #4-81         5-123        9-252
FF.MSK   = 000077           #4-81
FF.PRT   = 000200           #4-81         4-81
FF.QUL   = 000370           #4-81
FF.REM   = 100000           #4-81
FL.DLM   = ******  GX        11-340       11-366
FL.X25   = ******  GX        11-338
F.ADD      000016           #4-81
F.CEV      000030           #4-81
F.CIR      000024           #4-81
F.CLS      000002           #4-81
F.EVT      000004           #4-81
F.FLG      000014           #4-81
F.LEN      000016           #4-81
F.LENX   = ******  GX        9-293        9-294
F.LIN      000020           #4-81
F.LNK      000000           #4-81
F.MOD      000022           #4-81
F.REM      000026           #4-81
F.SEV      000004           #4-81
OBJLEN   = 000013           #6-134        8-197
OC.CTM   = ******  GX        11-361
OM.CON   = ******  GX        9-247
OM.FIL   = ******  GX        9-250
OM.KEV   = ******  GX        9-231        9-297
O.LENX   = ******  GX        8-200        8-201
PURCIR     000756 R          5-109        5-110       #11-330
PURLIN     000734 R          5-107       #10-311
```

```
    416 000000                                          STATE$  ,&P$DST
    417 000000                                           TRAN$  $STRNG, $EXIT,  .MODST
    418                                         ;
    419                                         ; DEFINE COMPONENTS
    420                                         ;
    421 000000                                          STATE$  DEFINE, &P$DEF
    422 000000                                           TRAN$  !DEFCIR,EXIT
    423 000000                                           TRAN$  !DEFLIN,EXIT
    424 000000                                           TRAN$  !DEFPRO,EXIT
    425 000000                                           TRAN$  !DEFEXE,EXIT
    426 000000                                           TRAN$  !DEFSYS,EXIT
    427 000000                                           TRAN$  !DEFOBJ,EXIT
    428 000000                                           TRAN$  !DEFNOD,EXIT
    429 000000                                           TRAN$  !DEFLOG,EXIT
    430 000000                                           TRAN$  !DEFMOD,EXIT
```

```
                                          .SBTTL  DEFINE PROCESS COMMAND
                                  ;
                                  ; DEFINE PROCESS COMMAND
                                  ;
 000000                                   STATE$  DEFPRO
 000000                                    TRAN$  !PROID,PROC,,CO.PRO,$COMP
 000000                                    TRAN$  !KNPRO,,,CO.KNP,$COMP

 000000                                   STATE$  PROC,&P$DEPR
 000000                                    TRAN$  !PROPT

 000000                                   STATE$  PROO
 000000                                    TRAN$  !PROPT,PROO
 000000                                    TRAN$  $LAMDA,$EXIT
                                  ;
                                  ; PROCESS OPTIONS
                                  ;
 000000                                   STATE$  PROPT
 000000                                    TRAN$  !PSTATE,$EXIT,,OP.STA,$OPRND
 000000                                    TRAN$  !LINMAX,$EXIT,,OP.NLT,$OPRND
 000000                                    TRAN$  !CNTMAX,$EXIT,,OP.NCT,$OPRND
                                  ;
                                  ; STATE
                                  ;
 000000                                   STATE$  PSTATE
 000000                                    TRAN$  &K$STA

 000000                                   STATE$  ,&P$PSTA
 000000                                    TRAN$  &K$ON,$EXIT,,SW.LOA,$SWICH
 000000                                    TRAN$  &K$CLE,$EXIT
                                  ;
                                  ; MAXIMUM LINES
                                  ;
 000000                                   STATE$  LINMAX
 000000                                    TRAN$  &K$MAX

 000000                                   STATE$  ,&P$PEN
 000000                                    TRAN$  &K$LIN

 000000                                   STATE$  ,&P$MXLI
 000000                                    TRAN$  $DNUMB,$EXIT,,PRNLT
                                  ;
                                  ; MAXIMUM CONTROLLERS
                                  ;
 000000                                   STATE$  CNTMAX
 000000                                    TRAN$  &K$MAX

 000000                                   STATE$  ,&P$PEN
 000000                                    TRAN$  &K$CON

 000000                                   STATE$  ,&P$MXCO
 000000                                    TRAN$  $DNUMB,$EXIT,,PRNCT
                                          .SBTTL  SUB-EXPRESSIONS FOR PROCESS ID
                                  ;
                                  ; LOOK FOR "KNOWN PROCESSES"
                                  ;
 000000                                   STATE$  KNPRO
```

```

     000000                          STATE$  ,&P$MNC
     000000                           TRAN$  $DNUMB, $EXIT,  .EXNC
                             ;
                             ; LOOK FOR ROUTING TIMER
                             ;
     000000                          STATE$  EROU
     000000                           TRAN$  &K$ROU
     000000                          STATE$  ,&P$ROU
     000000                           TRAN$  &K$TIM
     000000                          STATE$  ,&P$ROUT
     000000                           TRAN$  $DNUMB, $EXIT,  .EXRTM
                             ;
                             ; LOOK FOR EXECUTOR SUBADDRESS RANGE
                             ;
     000000                          STATE$  ESUB
     000000                           TRAN$  &K$SUBA
     000000                          STATE$  ,             &P$SUBA
     000000                           TRAN$  $DNUMB,,             .EXSUB
     000000                          STATE$
     000000                           TRAN$  <'->,    ESUB2
     000000                           TRAN$  $LAMDA, $EXIT
     000000                          STATE$  ESUB2
     000000                           TRAN$  $DNUMB, $EXIT,  .EXSB2


                             ;
                             ; Executor Maximum Area
                             ;
     000000                          STATE$  EAMAX
     000000                           TRAN$  &K$MAX
     000000                          STATE$  ,&P$MAX
     000000                           TRAN$  &K$ARE
     000000                          STATE$  ,&P$MXA
     000000                           TRAN$  $DNUMB, $EXIT,  .EAMAX
     000000                           TRAN$  $LAMDA, $FAIL
                             ;
                             ; Executor Area [Maximum] Hops
                             ;
     000000                          STATE$  EAHOPS
     000000                           TRAN$  &K$ARE
     000000                          STATE$  ,&P$AOP
     000000                           TRAN$  &K$MAX, AHOP2
     000000                           TRAN$  $LAMDA
     000000                          STATE$  AHOP2
     000000                           TRAN$  &K$HOP
     000000                          STATE$  ,&P$AHOP
     000000                           TRAN$  $DNUMB, $EXIT,  .EAHOP
     000000                           TRAN$  $LAMDA, $FAIL

                             ;
                             ; Executor Area [Maximum] Cost
                             ;
     000000                          STATE$ EACOST
     000000                           TRAN$  &K$ARE
     000000                          STATE$  ,&P$AOP
     000000                           TRAN$  &K$MAX, ACOS2
     000000                           TRAN$  $LAMDA
```

```
   2422 000000                                     TRAN$  <',>,    CHNL1
   2423 000000                                     TRAN$  <'->,    CHNL1,   .MOCRN
   2424 000000                                     TRAN$  $LAMDA, $EXIT
   2425
```

```
                                                                               N 11
CFEPRS      CREATED BY  MACRO  ON 2-JUL-85 AT 14:24        PAGE 4
MACRO CROSS REFERENCE                                      CREF   04.00

MACRO NAME      REFERENCES

                #12-2056  #12-2060  #12-2063  #12-2069  #12-2072  #12-2075  #12-2078  #12-2083  #12-2086  #12-2089
                #12-2092  #12-2097  #12-2100  #12-2103  #12-2106  #12-2111  #12-2114  #12-2117  #12-2120  #12-2125
                #12-2128  #12-2131  #12-2134   13-2141  #13-2151  #13-2154  #13-2160  #13-2163  #13-2166  #13-2172
                #13-2174  #13-2177  #13-2183  #13-2186  #13-2189  #13-2195  #13-2198  #13-2200  #13-2203  #13-2215
                #13-2217  #13-2221  #13-2237  #13-2240  #13-2242  #13-2246  #13-2256  #13-2259  #13-2261  #13-2265
                #13-2273  #13-2275  #13-2277  #13-2282  #13-2284  #13-2286  #13-2291  #13-2293  #13-2295  #13-2300
                #13-2302  #13-2304  #13-2309  #13-2311  #13-2313  #13-2319  #13-2321  #13-2323  #13-2328  #13-2330
                #13-2332  #13-2337  #13-2339  #13-2341  #13-2346  #13-2348  #13-2350  #13-2355  #13-2357  #13-2359
                #13-2365  #13-2367  #13-2369  #13-2375  #13-2377  #13-2379  #13-2384  #13-2386  #13-2388  #13-2393
                #13-2395  #13-2401  #13-2403  #13-2408  #13-2410  #13-2415  #13-2417  #13-2419  #13-2421   14-2431
                #14-2433  #14-2435  #14-2438  #14-2445  #14-2447  #14-2449  #14-2451  #14-2457  #14-2459  #14-2464
                #14-2466  #14-2471  #14-2473  #14-2475  #14-2480  #14-2482  #14-2484  #14-2489  #14-2491  #14-2494
                #14-2500  #14-2511  #14-2513  #14-2518  #14-2520  #14-2522  #14-2525  #14-2531  #14-2533  #14-2539
                #14-2541  #14-2547  #14-2549  #14-2555  #14-2557  #14-2563  #14-2568  #14-2571  #14-2574  #14-2576
                #14-2578  #14-2583  #14-2586  #14-2589  #14-2591  #14-2593  #14-2598  #14-2600  #14-2602  #14-2619
                #14-2623  #14-2626  #14-2629  #14-2632   15-2642  #15-2645
TRANS$          #4-60     #4-286    #4-287    #4-288    #4-289    #4-290    #4-291    #4-292    #4-293    #4-294
                #4-299    #4-300    #5-307    #5-308    #5-309    #5-310    #5-313    #5-314    #5-316    #5-317
                #5-320    #5-323    #5-324    #5-325    #5-326    #5-327    #5-328    #5-329    #5-330    #5-331
                #5-332    #5-333    #5-334    #5-337    #5-338    #5-344    #5-345    #5-346    #5-347    #5-352
                #5-353    #5-354    #5-355    #5-356    #5-357    #5-358    #5-363    #5-364    #5-365    #5-367
                #5-368    #5-369    #5-370    #5-375    #5-377    #5-378    #5-380    #5-381    #5-386    #5-388
                #5-393    #5-395    #5-400    #5-402    #5-408    #5-410    #5-415    #5-417    #5-422    #5-423
                #5-424    #5-425    #5-426    #5-427    #5-428    #5-429    #5-430    #6-437    #6-438    #6-439
                #6-440    #6-441    #6-442    #6-443    #6-444    #6-445    #6-446    #6-447    #6-448    #6-449
                #6-450    #6-451    #6-453    #6-458    #6-459    #6-465    #6-466    #6-468    #6-469    #6-470
                #6-471    #6-472    #6-473    #6-474    #6-475    #6-476    #6-477    #6-478    #6-479    #6-480
                #6-481    #6-486    #6-492    #6-493    #6-495    #6-501    #6-502    #6-504    #6-505    #6-507
                #6-513    #6-515    #6-521    #6-523    #6-528    #6-529    #6-531    #6-536    #6-538    #6-539
                #6-545    #6-546    #6-548    #6-553    #6-555    #6-556    #6-562    #6-564    #6-565    #6-571
                #6-572    #6-575    #6-576    #6-579    #6-585    #6-588    #6-589    #6-595    #6-598    #6-604
                #6-609    #6-614    #6-615    #6-620    #6-622    #6-627    #6-629    #6-630    #6-631    #6-632
                #6-634    #6-635    #6-640    #6-645    #6-647    #6-648    #6-653    #6-654    #6-661    #6-662
                #6-665    #6-666    #6-669    #6-670    #6-673    #6-674    #6-677    #6-680    #6-681    #6-684
                #6-685    #6-688    #6-691    #6-698    #6-700    #6-702    #6-704    #6-705    #6-707    #6-709
                #6-710    #6-711    #6-713    #6-715    #6-720    #6-721    #6-726    #6-728    #6-730    #6-735
                #6-742    #6-745    #6-746    #6-747    #6-752    #6-755    #7-763    #7-766    #7-767    #7-770
                #7-775    #7-777    #7-778    #7-779    #7-780    #7-781    #7-782    #7-783    #7-784    #7-785
                #7-786    #7-787    #7-788    #7-789    #7-790    #7-795    #7-797    #7-799    #7-800    #7-801
                #7-804    #7-805    #7-810    #7-811    #7-813    #7-815    #7-816    #7-819    #7-820    #7-821
                #7-822    #7-823    #7-824    #7-825    #7-826    #7-827    #7-828    #7-829    #7-830    #7-831
                #7-832    #7-833    #7-834    #7-835    #7-836    #7-837    #7-838    #7-839    #7-840    #7-841
                #7-842    #7-843    #7-844    #7-849    #7-850    #7-853    #7-854    #7-857    #7-858    #7-861
                #7-864    #7-865     7-868    #7-869    #7-870    #7-871    #7-872    #7-877    #7-879    #7-881
                #7-882    #7-887    #7-889    #7-891    #7-892    #7-897    #7-898    #7-900    #7-902    #7-904
                #7-905    #7-910    #7-912    #7-913    #7-915    #7-916    #7-919    #7-921    #7-922    #7-923
                #7-924    #7-930    #7-931    #7-933    #7-935    #7-936    #7-941    #7-944    #7-945    #7-948
                #7-950    #7-955    #7-958    #7-959    #7-962    #7-964    #7-969    #7-970    #7-973    #7-974
                #7-977    #7-980    #7-982    #7-987    #7-990    #7-991    #7-994    #7-996    #7-1001   #7-1004
                #7-1007   #7-1009   #7-1015   #7-1016   #7-1022   #7-1025   #7-1026   #7-1029   #7-1032   #7-1035
                #7-1036   #7-1039   #7-1044   #7-1046   #7-1049   #7-1050   #7-1053   #7-1054   #7-1057   #7-1058
                #7-1061   #7-1062   #7-1063   #7-1066   #7-1069   #7-1076   #7-1079   #8-1086   #8-1087   #8-1090
```

N 12

```
CFERQB - DYNAMIC BUFFER MANAGEM MACRO V05.03b Saturday 29-Jun-85 05:13
Table of contents
```

```
     4-   46     MACRO CALLS
     5-   59     $RQBF - REQUEST CORE BLOCK WITH TASK EXTENSION
     6-  111     .RQCB   REQUEST CORE BLOCK
     6-  112     $RLBF   RELEASE CORE BLOCK
```

```
                        ;****
                        ; LOCAL MACROS
                        ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT

                                .MACRO  ERROR$  TEXT
                        ;
                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                        ; MESSAGE STRING.
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                .ENDC
                                CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                .ENDM   ERROR$

                                .MACRO  EPRINT  TEXT
                        ;
                        ; PRINT TEXT ON ERROR LUN
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                .ENDC
                                CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                .ENDM   EPRINT

                        ;****
                        ; MACRO CALLS
                        ;****

                                .MCALL  ISTAT$,STATE$,TRAN$


                        ;****
                        ; LOCAL DATA
                        ;****

        000073          SEMI    = ';                    ; SEMICOLON FOR TPARS TABLES
        000076          RBRKT   = '>                    ; RIGHT BRACKET
000000                  $ERROR::.BLKW   1               ; NON-ZERO IF SYNTAX ERROR
```

```
;+
;       UTILITIES
;
; THIS FILE CONTAINS COMMONLY USED SUBROUTINES TO PERFORM
; REPEATED FUNCTIONS.
;
;       $PRINT - PRINT ROUTINE
;
;       $CFERR - PRINT ERROR MESSAGE AND SET UP EXIT STATUS
;       $CFER1 - JUST PRINT ERROR MESSAGE
;
;       $MVASC - MOVE ASCII STRING TO OUTPUT BUFFER
;
;       $QINS - INSERT ENTRY AT END OF QUEUE
;       $QINSN - INSERT ENTRY INTO MIDDLE OF QUEUE
;
;       $C5TA - CONVERT EXTENDED RAD50 TO ASCII AND REMOVE SPACES AT END
;       $C5TA1- CONVERT EXTENDED RAD50 ("-") TO ASCII, TRIMMED
;
;       $CAT5 - CONVERT ASCII TO RAD50
;       .CAT5 - CONVERT ASCII TO EXTENDED RAD50 ("*")
;       .CAT5E- CONVERT ASCII TO EXTENDED RAD50 ("-")
;
;       $FNPRO - FIND NEXT PROCESS IN TEMPLATES
;
;       $FNSLT - FIND NEXT SLT IN TEMPLATES
;       .FNSLT - FIND NEXT SLT FOR SET KNOWN LINES/CIRCUITS
;
;       $FNSTA - FIND NEXT STATION IN TEMPLATES
;
;       $UPBLK - UPDATE POOL BLOCK ALLOCATION
;
;-
```

```
                                         .SBTTL  $UPBLK - UPDATE POOL BLOCK ALLOCATION
                                 ;+
                                 ; *** $UPBLK - UPDATE POOL BLOCK ALLOCATION
                                 ;
                                 ; INPUTS        TEMPLATES IN LIST:
                                 ;                       LLC$DF FOR XPT (NODE COUNTERS)
                                 ;                       ROU$DF (MAX ADDRESS, OPTIONAL)
                                 ;                       $PBLK - POLL BLOCK AREA FUDGE FACTOR
                                 ;
                                 ; OUTPUTS       CARRY CLEAR:
                                 ;                       PAR$DF BLOCK ALLOCATION UPDATED.
                                 ;               CARRY SET:
                                 ;                       ERROR - DEC$DF OR PAR$DF MISSING
                                 ;
                                 ; REGISTERS     ALL REGISTERS ARE PRESERVED
                                 ;
                                 ;-

001244                           $UPBLK::
001244  004767  000000G                  CALL    $SAVAL          ; SAVE REGISTERS
001250  016700  000000G                  MOV     $TMLST,R0       ; POINT TO TEMPLATE LISTHEAD
001254  005001                           CLR     R1              ; NO NODE COUNTERS AS YET
001256  005002                           CLR     R2              ; NO PARTITION TEMPLATE YET
001260  005004                           CLR     R4              ; NO MAX ADDRESS AS YET

001262  011000                   10$:    MOV     (R0),R0         ; NEXT TEMPLATE
001264  001423                           BEQ     40$             ; IF EQ, DONE WITH LIST
001266  126027  000000G 000000G          CMPB    C.STS(R0),#CS.PAR ; CHECK FOR PARTITION TEMPLATE
001274  001002                           BNE     20$             ; BR IF NE
001276  010002                           MOV     R0,R2           ; ELSE SAVE ADDRESS
001300  000770                           BR      10$             ; CONTINUE
001302  126027  000000G 000000G  20$:    CMPB    C.STS(R0),#CS.DEC ; CHECK FOR DEC$DF TEMPLATE
001310  001002                           BNE     30$             ; BR IF NE
001312  010001                           MOV     R0,R1           ; ELSE SAVE ADDRESS
001314  000762                           BR      10$             ; CONTINUE
001316  126027  000000G 000000G  30$:    CMPB    C.STS(R0),#CS.ROU ; CHECK FOR ROUTING TEMPLATE
001324  001356                           BNE     10$             ; BR IF NE
001326  016004  000000G                  MOV     C.PRM+0(R0),R4  ; ELSE SAVE MAX ADDRESS
001332  000753                           BR      10$

001334  005702                   40$:    TST     R2              ; GOT A PARTITION TEMPLATE ?
001336  001444                           BEQ     50$             ; IF EQ, NO - ERROR
001340  005701                           TST     R1              ; GOT XPT LLC TEMPLATE ?
001342  001442                           BEQ     50$             ; IF EQ, NO - ERROR
001344  016101  000006                   MOV     C.NDC(R1),R1    ; ELSE GET NUMBER OF NODE COUNTERS
001350  012700  000050                   MOV     #E$NLEN,R0      ; SIZE OF A COUNTER BLOCK
001354  004767  000000G                  CALL    $MUL            ; CALCULATE BYTES FOR NODE COUNTERS
001360  006304                           ASL     R4              ; 2 BYTES PER NODE FOR ROUTING TABLES
001362  060401                           ADD     R4,R1           ; ADD TO ALLOCATION SO FAR
001364  062701  000077                   ADD     #63.,R1         ; ALWAYS ROUND TO NEXT BLOCK
        000006                           .REPT   6
                                         ASR     R1              ; CONVERT TO BLOCKS
                                         .ENDR
001404  016700  000000G                  MOV     $PBLK,R0        ; GET FUDGE FACTOR
001410  100010                           BPL     45$             ; IF PL, GOT ONE
001412  016200  000010G                  MOV     C.PRM+10(R2),R0 ; ELSE GET BLOCK ALLOCATION
001416  160100                           SUB     R1,R0           ; CALCULATE FUDGE FACTOR IN BLOCKS
```

```
B  1              J  5              E 10              M 14
C  1              K  5              F 10              N 14
D  1              L  5  CFEPRG      G 10              B 15
E  1              M  5              H 10              C 15
F  1              N  5              I 10              D 15
G  1              B  6              J 10              E 15
H  1              C  6              K 10              F 15
I  1              D  6              L 10              G 15
J  1              E  6              M 10              H 15
K  1              F  6              N 10              I 15
L  1              G  6              B 11              J 15
M  1              H  6              C 11              K 15
N  1              I  6              D 11              L 15
B  2              J  6              E 11              M 15
C  2              K  6              F 11              N 15
D  2              L  6              G 11              B 16
E  2              M  6              H 11              C 16
F  2              N  6              I 11              D 16
G  2              B  7              J 11              E 16
H  2              C  7              K 11              F 16
I  2              D  7  CFEPRS      L 11              G 16
J  2              E  7              M 11              H 16
K  2              F  7              N 11              I 16
L  2              G  7              B 12              J 16
M  2              H  7              C 12              K 16
N  2              I  7              D 12  CFEROT      L 16  CFETSB
B  3              J  7              E 12              M 16
C  3              K  7              F 12
D  3              L  7              G 12
E  3              M  7              H 12
F  3              N  7              I 12
G  3              B  8              J 12
H  3              C  8              K 12
I  3              D  8              L 12
J  3              E  8              M 12  CFERQB
K  3              F  8              N 12
L  3  CFEINI      G  8              B 13
M  3              H  8              C 13
N  3              I  8              D 13
B  4              J  8              E 13
C  4              K  8              F 13
D  4              L  8              G 13
E  4              M  8              H 13
F  4              N  8              I 13
G  4  CFELIS      B  9              J 13
H  4              C  9              K 13  CFESTB
I  4              D  9              L 13
J  4              E  9              M 13
K  4              F  9              N 13
L  4              G  9              B 14
M  4              H  9              C 14
N  4              I  9              D 14
B  5  CFEOPN      J  9              E 14
C  5              K  9              F 14
D  5              L  9              G 14
E  5              M  9              H 14
F  5              N  9              I 14  CFESUB
G  5              B 10              J 14
H  5              C 10              K 14
I  5              D 10              L 14
```

```
                        .TITLE  CFETSB - TEMPLATE PARSE SUBROUTINES
                        .IDENT  /V05.00/
                        .ENABL  LC

                ;
                ; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ;       CETAB TEMPLATE TEMPLATE PARSE SUBROUTINES
                ;
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-AUG-81
                ;       DECNET-11M/S V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                                   .SBTTL  $CFUPD - UPDATE THE CETAB FILE
                                          ;+
                                          ; *** - $CFUPD - UPDATE THE CETAB FILE
                                          ;
                                          ; INPUT:
                                          ;        NONE.
                                          ;
                                          ; OUTPUT:
                                          ;        C-BIT = SUCCESS/FAILURE
                                          ;        THE CETAB FILE IS UPDATED.
                                          ;
                                          ;-
 000170                                   $CFUPD::
 000170   005067  177604                         CLR     $ERROR            ; ERROR INDICATOR
 000174   010346                                 MOV     R3,-(SP)          ; SAVE R3
 000176   012700  000000G                        MOV     #$IFDB,R0         ; POINT AT INPUT FILE FDB
 000202   016701  000000G                        MOV     $HIVBN,R1         ; POISTION NEXT READ FROM START OF FILE
 000206   016702  000000G                        MOV     $LOVBN,R2         ;      ...
 000212   016703  000000G                        MOV     $BYTE,R3          ;      ...
 000216   004767  000000G                        CALL    .POINT            ; POINT TO START OF FILE
 000222   012603                                 MOV     (SP)+,R3          ; RESTORE R3
 000224   103006                                 BCC     10$               ; IF CC, ALL OKAY
 000226                                          ERROR$  #ERRFIL           ; ELSE, ERROR OPENING FILE
 000236   000261                                 SEC                       ; HARD FAILURE
 000240   000472                                 BR      80$               ; AND RETURN
 000242   004767  000000G                 10$:   CALL    $RDATA            ; READ A RECORD
 000246   103466                                 BCS     70$               ; IF CS, END OF FILE
 000250   010367  177526                         MOV     R3,$LEN           ; SAVE LENGTH OF RECORD
 000254   012700  000000G                        MOV     #$OFDB,R0         ; POINT AT OUTPUT FILE FDB
 000260   012760  000000G 000002G                MOV     #$BUFR,F.NRBD+2(R0) ; ASSUME $BUFR IS THE OUTPUT BUFFER
 000266   005001                                 CLR     R1                ; TPARS INITIALIZATION - EXACT MATCH
 000270   012702  000000'                        MOV     #UPDKTB,R2        ; KEYWORD TABLE
 000274   012704  000000G                        MOV     #$BUFI,R4         ; ADDRESS OF RECORD
 000300   012705  000000'                        MOV     #UPDSTB,R5        ; STATE TABLE
 000304   004767  000000G                        CALL    .TPARS            ; PARSE THE RECORD - IGNORE ERRORS
 000310   103403                                 BCS     20$               ; IF CS, ERROR
 000312   005767  177462                         TST     $ERROR            ; SYNTAX ERROR ?
 000316   001425                                 BEQ     60$               ; IF EQ, NO
 000320                                   20$:   ERROR$  #ERRSYN           ; SYNTAX ERROR

                                          ;
                                          ; INSERT ERROR FLAG IN OFFENDING LINE AND PRINT IT
                                          ;
 000330   010400                          30$:   MOV     R4,R0             ; COPY ADDRESS OF OFFENDING SYMBOL
 000332   105720                          40$:   TSTB    (R0)+             ; LOCATE END OF STRING
 000334   001376                                 BNE     40$               ; IF NE, MORE
 000336   114060  000002                  50$:   MOVB    -(R0),2(R0)       ; SLIDE EVERYTHING DOWN 2 BYTES
 000342   020004                                 CMP     R0,R4             ; HAVE WE REACHED START OF ERROR ?
 000344   001374                                 BNE     50$               ; IF NE, NO
 000346   112720  000076                         MOVB    #'>,(R0)+         ; INSERT ERROR FLAG
 000352   112720  000076                         MOVB    #'>,(R0)+         ;   ...
 000356                                          EPRINT  #$BUFI            ; PRINT THE BAD LINE
 000366   000261                                 SEC                       ; SHOW FATAL ERROR
 000370   000416                                 BR      80$               ; AND LEAVE
 000372   016767  177404  000000C 60$:   MOV     $LEN,F.NRBD+$OFDB ; SET LENGTH OF BUFFER
 000400   001720                                 BEQ     10$               ; IF EQ, NOTHING IN LINE
```

```
                                                                                B 3
**FILE**ID**CFEUSB




  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   UU      UU     SSSSSSSS   BBBBBBBB
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   UU      UU     SSSSSSSS   BBBBBBBB
CC           FF           EE           UU      UU   SS           BB      BB
CC           FF           EE           UU      UU   SS           BB      BB
CC           FF           EE           UU      UU   SS           BB      BB
CC           FF           EE           UU      UU   SS           BB      BB
CC           FFFFFFFF     EEEEEEEE     UU      UU     SSSSSS     BBBBBBBB
CC           FFFFFFFF     EEEEEEEE     UU      UU     SSSSSS     BBBBBBBB
CC           FF           EE           UU      UU           SS   BB      BB
CC           FF           EE           UU      UU           SS   BB      BB
CC           FF           EE           UU      UU           SS   BB      BB         ....
CC           FF           EE           UU      UU           SS   BB      BB         ....
  CCCCCCCC   FF           EEEEEEEEEE   UUUUUUUUUU   SSSSSSSS     BBBBBBBB           ....
  CCCCCCCC   FF           EEEEEEEEEE   UUUUUUUUUU   SSSSSSSS     BBBBBBBB           ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                                C 3
CFEUSB - TEMPLATE UPDATE SUBROU MACRO V05.03b Saturday 29-Jun-85 05:16
Table of contents
```

```
     58                                                     .ENDM
     59
     60                                                     .MACRO  EVENT   CLASS,EVNUM
     61                                                     $$$W0=0
     62                                                     $$$W1=0
     63                                                     $$$W2=0
     64                                                     $$$W3=0
     65                                                     .IRP    ...N, <EVNUM>
     66                                                     ...BIS  \...N/16. \...N&15
     67                                                     .ENDR
     68                                                     .RADIX  8
     69                                                     .WORD   ^O<100>*^D<CLASS>,$$$W0,$$$W1,$$$W2,$$$W3
     70                                                     .RADIX  10
     71                                                     .ENDM
     72
     73                                                     .SBTTL  Known events table
     74                                             ;****
     75                                             ; Local data
     76                                             ;****
     77
     78                                             ;
     79                                             ; Known events table:
     80                                             ;
     81                                             ;       Each entry in this table consists one word specifying
     82                                             ;       the event class followed by 4 words of event masks. Note
     83                                             ;       that the radix for events and classes is DECIMAL.
     84                                             ;
     85                                                     .LIST   MEB
     86 000000                                      $KNEVT::
     87
     88 000000                                              EVENT   0,<0,1,2,3,4,5,6,7,8,9>
        000000  000000  001777  000000                      .WORD   ^O<100>*^D<0>,$$$W0,$$$W1,$$$W2,$$$W3
        000006  000000  000000
     89 000012                                              EVENT   2,<0,1>
        000012  000200  000003  000000                      .WORD   ^O<100>*^D<2>,$$$W0,$$$W1,$$$W2,$$$W3
        000020  000000  000000
     90 000024                                              EVENT   3,<0,1,2>
        000024  000300  000007  000000                      .WORD   ^O<100>*^D<3>,$$$W0,$$$W1,$$$W2,$$$W3
        000032  000000  000000
     91 000036                                              EVENT   4,<0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19>
        000036  000400  177777  000017                      .WORD   ^O<100>*^D<4>,$$$W0,$$$W1,$$$W2,$$$W3
        000044  000000  000000
     92 000050                                              EVENT   5,<0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18>
        000050  000500  177777  000007                      .WORD   ^O<100>*^D<5>,$$$W0,$$$W1,$$$W2,$$$W3
        000056  000000  000000
     93 000062                                              EVENT   6,<0,1,2,3,4,5>
        000062  000600  000077  000000                      .WORD   ^O<100>*^D<6>,$$$W0,$$$W1,$$$W2,$$$W3
        000070  000000  000000
     94 000074                                              EVENT   64,<0,1,2>
        000074  010000  000007  000000                      .WORD   ^O<100>*^D<64>,$$$W0,$$$W1,$$$W2,$$$W3
        000102  000000  000000
     95 000106                                              EVENT   65,<0>
        000106  010100  000001  000000                      .WORD   ^O<100>*^D<65>,$$$W0,$$$W1,$$$W2,$$$W3
        000114  000000  000000
     96 000120                                              EVENT   66,<0>
        000120  010200  000001  000000                      .WORD   ^O<100>*^D<66>,$$$W0,$$$W1,$$$W2,$$$W3
        000126  000000  000000
        .IF NDF $GOFF
        .TITLE  PRMPTS - PROMPTS  STRING ASSEMBLY
        .IFF
        .TITLE  PRMPTO - PROMPTS, OFFSET ASSEMBLY
        .ENDC

        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1984, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       NETWORK MANAGEMENT - PROMPTS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 2.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;


```

```
     59                                              .SBTTL  MACROS AND SYMBOLS AND INITIALIZATION
     60                                      ;
     61                                      ; DECLARE EXTERNAL MACROS
     62                                      ;
     63                                              .MCALL  ISTAT$,PRMPT$
     64                                      ;
     65                                      ; SIGNAL PROMPTING SUPPORT
     66                                      ;
     67           000001                             $GPRM   = 1
     68                                      ;
     69                                      ; MAKE TABLES READ ONLY
     70                                      ;
     71           000001                             $RONLY  = 1
```

```
                                                .SBTTL  $BL.MAN - MAKE BLANK PROCESSING MANUAL
                                        ;+
                                        ; *** - $BL.MAN - MAKE BLANK PROCESSING MANUAL
                                        ;
                                        ; INPUTS:
                                        ;       NONE.
                                        ;
                                        ; OUTPUTS:
                                        ;       .PFLAG = MODIFIED
                                        ;
                                        ;-
000150                                  $BL.MN1::
000150  112767  000001  000001G                 MOVB    #1,.PFLAG+1             ; ALLOW ONE CHARACTER ABBREVIATIONS
000156                                  $BL.MAN::
000156  052767  000001  000000G                 BIS     #1,.PFLAG               ; DON'T IGNORE BLANKS
000164  016767  000000G 000000G                 MOV     .PSTPT,START            ; SAVE START ADDRESS OF STRING
000172  000207                                  RETURN
```

```
    483                                         .SBTTL  .PRMDA - PARSE MAXIMUM DATA
    484                                 ;+
    485                                 ; ** .PRMDA - PARSE MAXIMUM DATA
    486                                 ;
    487                                 ; INPUTS         .TPARS VARIABLES.
    488                                 ;
    489                                 ; OUTPUT         $BSZ = PARSED MAXIMUM DATA
    490                                 ;                THE TRANSITION IS REJECTED IF THE VALUE - 5
    491                                 ;                IS NOT A POWER OF 2.
    492                                 ;
    493                                 ;-
    494 001322                          .PRMDA::
    495 001322  005767  000000G                 TST     .PNUMH          ; TEST HIGH ORDER
    496 001326  001236                          BNE     ILLNUM          ; IF NE, OUT OF RANGE
    497 001330  016701  000000G                 MOV     .PNUMB,R1       ; GET THE VALUE
    498 001334  010167  000000G                 MOV     R1,$BSZ         ; STUFF INTO PARSED BLOCK SIZE AREA
    499 001340  162701  000005                  SUB     #5,R1           ; SUBTRACT 5
    500 001344  020127  002000                  CMP     R1,#1024.       ; CHECK RANGE
    501 001350  101225                          BHI     ILLNUM          ; IF HI, TOO HIGH
    502 001352  012700  000010                  MOV     #8.,R0          ; LOW RANGE /2
    503 001356  006300                  10$:    ASL     R0              ; NEXT POWER OF 2
    504 001360  001621                          BEQ     ILLNUM          ; IF EQ, NOT A POWER OF 2
    505 001362  020001                          CMP     R0,R1           ; CHECK FOR POWER OF 2
    506 001364  001374                          BNE     10$             ; IF NE, KEEP LOOKING
    507 001366  000207                          RETURN
    508
```

```
                                     ;
                                     ; Check for PSI circuit
                                     ;
000024  005777  000000G                      TST     @PSIPT                  ; Check PSI present             ; RJK05
000030  001416                               BEQ     15$                     ; BR if PSI not loaded
000032                                       CALL    STOCIR                  ; Store circuit-id in context area
000036                                       CALL    $FNPVC                  ; Look for PSI circuit
000042  103156                               BCC     120$                    ; BR if found one
000044                                       CALL    NEWPVC                  ; Check for new X.25 circuit
000050  103153                               BCC     120$                    ; BR if found one
000052  010504                               MOV     R5,R4                   ; Clear context area again
000054  012700  000026                       MOV     #<L$PDV/2>,R0           ; Set length of buffer to clear
000060  005024                       12$:    CLR     (R4)+                   ; ...
000062                                       SOB     R0,12$                  ; ...

                                             .ENDC   ; NDF R$$11S & R$$PRO

000066  112104                       15$:    MOVB    (R1)+,R4                ; GET LENGTH OF STRING
000070  010100                               MOV     R1,R0                   ; COPY ADDRESS OF STRING
000072  060004                               ADD     R0,R4                   ; COMPUTE THE ADDRESS OF THE STRING END
000074  010466  000000G                      MOV     R4,R$R1(SP)             ; RETURN END OF STRING IN SAVED R1
                                     ;
                                     ; PARSE DEVICE NAME
                                     ;
000100  112703  000055                       MOVB    #'-,R3                  ; SET UP ALLOWED TERMINATOR
000104                               20$:    CALL    CAT5                    ; CONVERT PROCESS NAME TO RAD50
000110  103527                               BCS     110$                    ; IF CS, ILLEGAL CHARACTER - ERROR
000112  010165  000000                       MOV     R1,L$DDM(R5)            ; STORE RAD50 NAME
                                     ;
                                     ; PARSE CONTROLLER NUMBER
                                     ;
000116  005702                               TST     R2                      ; END OF THE STRING ?
000120  001123                               BNE     110$                    ; IF NE, YES - ERROR (NO CONTROLLER)
000122                                       CALL    CHKMUX                  ; CHECK IF DDM IS A MUX DEVICE
000126  132765  000004  000041               BITB    #LP$MUX,L$PFG(R5)       ; IS IT A MUX DEVICE ? ('-' is needed)
000134  001002                               BNE     30$                     ; IF NE, YES - CONTINUE
000136  112703  000056                       MOVB    #'.,R3                  ; ELSE, NO - SET NEW LEGAL TERMINATOR
000142                               30$:    CALL    CHKWLD                  ; IS CONTROLLER WILD ?
000146  103404                               BCS     40$                     ; IF CS, NO
000150  152765  000040  000041               BISB    #LP$WCN,L$PFG(R5)       ; ELSE, REMEMBER WILD CONTROLLER
000156  000407                               BR      50$                     ; AND CONTINUE
000160  005702                       40$:    TST     R2                      ; IF NE, Wild card in bad position
000162  001102                               BNE     110$                    ; BR IF error
000164                                       CALL    CDTB                    ; CONVERT CONTROLLER NUMBER
000170  103477                               BCS     110$                    ; IF CS, ERROR
000172  110165  000002                       MOVB    R1,L$CTL(R5)            ; else, STORE controller number
000176  132765  000004  000041       50$:    BITB    #LP$MUX,L$PFG(R5)       ; IS THIS A MUX DEVICE ?
000204  001431                               BEQ     70$                     ; IF EQ, NO - LOOK FOR TRIBUTARY NUMBER
                                     ;
                                     ; PARSE UNIT NUMBER
                                     ;
000206  005702                               TST     R2                      ; END OF STRING ?
000210  001067                               BNE     110$                    ; IF NE, YES - ERROR (NO UNIT NUMBER)
000212  012703  000056                       MOV     #'.,R3                  ; SET UP ALLOWED TERMINATOR
000216                                       CALL    CHKWLD                  ; CHECK IF WILD UNIT #
000222  103404                               BCS     60$                     ; IF CC, NO
000224  152765  000101  000041               BISB    #LP$WUN!LP$UNT,L$PFG(R5); ELSE, REMEMBER WILD UNIT NUMBER
```

```
                                                                                B 10
PRSCIR - PARSE A CIRCUIT-ID     MACRO V05.03b Saturday 29-Jun-85 13:20  Page 16-1
Symbol table

A$$CHK= 000000      HF$DLM= 000002      LF.PAC= 000200      L$SLT   000042      MC$LDR  001762
A$$CPS= 000000      HF$GWY= 000010      LF.RDY= 040000      L$SNM   000046      MC$LDS  001763
A$$PRI= 000000      HF$HOS= 000004      LF.REA= 010000      L$SYL   000122      MC$LLB  002021
A$$TRP= 000000      HF$XDF= 000020      LF.SER= 000040      L$TNM   000052      MC$LLP  002115
BIT0  = 000001      H$CUG   000010      LF.TIM= 000010      L$TPT   000050      MC$LLR  002007
BIT1  = 000002      H$DST   000012      LF.UNL= 020000      L$TRB   000040      MC$LOO  000001
BIT10 = 002000      H$D29   000014      LF.X2P= 000000      L$TSZ   000070      MC$LRB  002020
BIT11 = 004000      H$FLG   000000      LN.CLO= 000000      L$TYP   000063      MC$LRP  002114
BIT12 = 010000      H$GLEN  000104      LN.DUM= 000005      L$UNT   000003      MC$LRR  002006
BIT13 = 020000      H$GLT   000044      LN.LOA= 000004      L$$ASG= 000000      MC$LSI  002032
BIT14 = 040000      H$GNAM  000050      LN.LOO= 000003      L$$DRV= 000000      MC$LST  002033
BIT15 = 100000      H$GNML= 000020      LN.OAU= 000003      L$$P11= 000001      MC$LUP  006072
BIT2  = 000004      H$GPT   000046      LN.OFF= 000001      L$$11R= 000000      MC$MAP  010000
BIT3  = 000010      H$HITS  000034      LN.ON = 000000      L.COST  000015      MC$MBL  001764
BIT4  = 000020      H$HLEN  000044      LN.OOP= 000004      L.CTL   000012      MC$MBX  004420
BIT5  = 000040      H$LBDA  000070      LN.OPE= 000001      L.CVA   177776      MC$MBY  001752
BIT6  = 000100      H$LBDN  000072      LN.REF= 000002      L.DDM   000002      MC$MPX  004406
BIT7  = 000200      H$LDTE  000002      LN.SER= 000002      L.DDS   000004      MC$NAP  001604
BIT8  = 000400      H$LEN   000042      LN.STA= 000017      L.DLC   000003      MC$NBR  001130
BIT9  = 001000      H$LOTS  000032      LN.SUB= 000360      L.DLM   000006      MC$NBS  001131
B$10    000003      H$NETW  000024      LN.TRI= 000006      L.DLS   000010      MC$NCR  001154
B$11    000004      H$NML = 000006      LP$MPT= 000010      L.FLG   000000      MC$NCS  001155
B$12    000005      H$NPT   000022      LP$MUX= 000004      L.KRBA  000016      MC$NML  001274
B$5     000000      H$PTB   000020      LP$TRB= 000002      L.LEN = 000022      MC$NMR  001142
B$6     000001      H$PVC   000006      LP$UNT= 000001      L.MPF   000022      MC$NMS  001143
B$7     000002      H$RDTE  000004      LP$WCN= 000040      L.NMST  000020      MC$NNO  001606
B$7S10  000006      H$RNW   000042      LP$WDV= 000020      L.NSTA  000014      MC$NNU  001605
CAT5    000564R     H$SVC   000036      LP$WLD= 000360      L.OWNR  000021      MC$NOP  001607
CDTB    000706R     H$TRB   000016      LP$WTR= 000200      L.UNT   000013      MC$NOR  000000
CF$BLK= 000102      H$XAVL  000100      LP$WUN= 000100      MB$DIS  000001      MC$NPF  001616
CHKDEV  001036R     H$XBIA  000074      L$ADJ   000110      MB$ENA  000000      MC$NPR  001630
CHKMUX  001004R     H$X29C  000040      L$BUF   000060      MB$FUL  000002      MC$NRE  001200
CHKWLD  001372R     I$$RAR= 000000      L$CHN   000054      MB$MIX  000002      MC$NRT  001166
CVTBL   000556R     I$$RDN= 000000      L$COU   000114      MB$ONE  000001      MC$NVR  001642
C$DTE   000012      K$$CNT= 177546      L$CTB   000053      MB$RXO  000001      MC$P08  000000
C$FLG   000015      K$$CSR= 177546      L$CTL   000002      MB$TXO  000000      MC$P11  000001
C$GNAM  000016      K$$LDC= 000000      L$DDM   000000      MB$ZER  000000      MC$RFL  002046
C$LCN   000010      K$$TPS= 000074      L$FLG   000064      MC$BAB  006073      MC$SBU  002051
C$LEN   000016      LC$EXT  000000      L$FLX   000070      MC$BID  001765      MC$SFL  002044
C$LNK   000000      LC$INT  000001      L$LEN   000124      MC$BMC  001767      MC$SYC  000012
C$NAM   000002      LD$LP = 000000      L$LTM   000070      MC$BSC  001766      MC$SYL  000036
C$NML   000002      LF$END= 000020      L$MSG   000056      MC$CAC  001442      MC$SYR  000050
C$PORT  000014      LF$MLT= 040000      L$MTYP  000067      MC$CAP  001440      MC$SYS  000024
C$$CKP= 000000      LF$REA= 000001      L$NAM   000000      MC$CCL  001445      MC$SYZ  000000
C$$ORE= 000400      LF$SEG= 100000      L$NLEN= 000020      MC$CDC  002045      MC$TBR  001140
C$$RSH= 177564      LF$SIG= 000040      L$NCD   000112      MC$CDP  001441      MC$TBS  001141
D$$BUG= 177514      LF$SKP= 000004      L$NXT   000044      MC$CIF  001465      MC$TDR  002053
D$$ISK= 000000      LF$VR2= 000010      L$OPT   000062      MC$CLD  001464      MC$TIM  000000
D$$L11= 000001      LF$ZER= 000002      L$PAR   000120      MC$COU  100000      MC$TYP  007777
D$$YNC= 000000      LF.ACT= 100000      L$PDV   000055      MC$CTL  001454      MC$UBU  002052
D$$YNM= 000000      LF.BRO= 000400      L$PFG   000041      MC$CTR  001452      MC$UFD  002047
E$$XPR= 000000      LF.BWT= 000007      L$PLB   000116      MC$CTS  001453      MC$UMR  001132
F$$LVL= 000001      LF.ENA= 002000      L$PRO   000067      MC$DOV  002050      MC$UMS  001133
G$$TPP= 000000      LF.LPB= 001000      L$PVC   000004      MC$LBR  001750      MC$VAX  000003
G$$TSS= 000000      LF.MDC= 000100      L$SCN   000020      MC$LBS  001751      MC$WID  060000
G$$TTK= 000000      LF.MFL= 004000      L$SCR   000072      MC$LDI  001774      MC$WIH  040000
G$$wRD= 000000      LF.MTP= 000020      L$SLEN  000066      MC$L.   001775      MC$WIL  020000
```

```
                                                                                C 10
PRSCIR - PARSE A CIRCUIT-ID     MACRO V05.03b Saturday 29-Jun-85 13:20  Page 16-2
Symbol table
```

```
                                          .SBTTL  .EXID - STORE EXECUTOR ID - QUOTED
                                          .SBTTL  .EXID1 - STORE EXECUTOR ID NON QUOTED
                                 ;+
                                 ; *** - .EXID - STORE EXECUTOR ID - QUOTED
                                 ; *** - .EXID1 - STORE EXECUTOR ID - NON QUOTED
                                 ;
                                 ; INPUT:
                                 ;        .PSTPT = ADDRESS OF ID STRING
                                 ;        .PSTCN = COUNT OF THE CHARS IN THE ID STRING
                                 ;
                                 ; OUTPUT:
                                 ;        $ID = EXECUTOR ID STRING
                                 ;        THE TRANSITION IS ACCEPTED OR REJE TED
                                 ;
                                 ;-

000000                           .EXID::
000000  005267  000000G                   INC     .PSTPT            ; SKIP FIRST QUOTE
000004  162767  000002  000000G           SUB     #2,.PSTCN         ; SUBTRACT END QUOTES FROM LENGTH
000012  016701  000000G                   MOV     .PSTPT,R1         ; GET START OF STRING
000016  016702  000000G                   MOV     .PSTCN,R2         ; AND LENGTH

000022  005302                   1$:      DEC     R2                ; ANY CHARS LEFT ?
000024  002417                            BLT     .EXID1            ; IF LT, NO

000026  122127  000042                    CMPB    (R1)+,#<''>       ; IS CURRENT CHARACTER A QUOTE ?
000032  001373                            BNE     1$                ; IF NE, NO
000034  005367  000000G                   DEC     .PSTCN            ; DECREMENT COUNT
000040  005302                            DEC     R2                ; SKIP QUOTE
000042  010146                            MOV     R1,-(SP)          ; SAVE R1
000044  010246                            MOV     R2,-(SP)          ; AND R2
000046  010100                            MOV     R1,R0             ; COPY BUFFER ADDRESS
000050  005201                            INC     R1                ; MOVE END OF STRING OVER SECOND QUOTE
000052  004767  000000G                   CALL    $MVASC            ; ...
000056  012602                            MOV     (SP)+,R2          ; RESTORE  R2
000060  012601                            MOV     (SP)+,R1          ; AND R1
000062  000757                            BR      1$                ; AND CONTINUE

000064                           .EXID1::
000064  042767  000001  000000G           BIC     #1,.PFLAG         ; IGNORE BLANKS AGAIN
000072  012700  000000G                   MOV     #$ID,R0           ; GET START OF ID STRING OUTPUT BUFFER
000076  012701  000040                    MOV     #STRMX,R1         ; AND LENGTH OF BUFFER
000102  112720  000040           5$:      MOVB    #' ,(R0)+         ; BLANK OUT THE ID BUFFER
000106  005301                            DEC     R1                ; REPEAT R1 TIMES...
000110  003374                            BGT     5$                ;      ...
000112  016701  000000G                   MOV     .PSTPT,R1         ; GET START ADDRESS OF STRING
000116  016702  000000G                   MOV     .PSTCN,R2         ; GET END OF STRING
000122  022702  000040                    CMP     #STRMX,R2         ; IS STRING TOO LONG ?
000126  103003                            BHIS    10$               ; IF HIS, NO - OKAY
000130  062716  000002                    ADD     #2,(SP)           ; ELSE, REJECT THE TRANSITION
000134  000405                            BR      20$               ; AND LEAVE
000136  012700  000000G          10$:     MOV     #$ID,R0           ; GET ADDRESS OF OUTPUT BUFFER
000142  010220                            MOV     R2,(R0)+          ; STORE THE STRING LENGTH
000144  004767  000000G                   CALL    $MVASC            ; MOVE THE TEXT INTO THE OUTPUT BUFFER
000150  000207                   20$:     RETURN                    ; AND LEAVE
```

```
                                                .SBTTL  .EXINT - EXECUTOR INCOMING TIMER
                                        ;+
                                        ; **    .EXINT - EXECUTOR INCOMING TIMER
                                        ;
                                        ; INPUTS:       TPARS CONVENTIONS.
                                        ;
                                        ; OUTPUT:       $EINC   = PARSED EXECUTOR INCOMING TIMER
                                        ;-
001136                                  .EXINT::
001136  116767  000000G 000000G                 MOVB    .PNUMB,$EINC    ; MOVE IT
001144  001406                                  BEQ     10$             ; IF EQ, ERROR
001146  005767  000000G                         TST     .PNUMH          ; TOO HIGH ?
001152  001003                                  BNE     10$             ; IF EQ, OK
001154  105767  000001G                         TSTB    .PNUMB+1        ; BYTE VALUE?
001160  001405                                  BEQ     20$             ; BR IF YES - OK

001162  012767  000000G 000000G 10$:            MOV     #ERRILN,$ERROR  ; ERROR IN NUMBER
001170  062716  000002                          ADD     #2,(SP)         ; ELSE REJECT TRANSITION

001174  000207                  20$:            RETURN
```

```
                                                .SBTTL  PRSLIN - PARSE AN ASCII LINE-ID
                                        ;+
                                        ; **-PRSLIN-PARSE AN ASCII LINE-ID
                                        ;
                                        ; THIS ROUTINE PARSES AN ASCII IMAGE LINE-ID STRING INTO
                                        ; DEVICE, CONTROLLE AND UNIT. WILD CARDS  ARE ALLOWED IN CONTROLLER
                                        ; AND UNIT FIELDS.
                                        ;
                                        ; INPUTS:
                                        ;       R1 = ADDRESS OF IMAGE LINE-ID STRING
                                        ;       R4 = ADDRESS OF A 7 WORD OUTPUT PARAMETER BLOCK
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT CLEAR:
                                        ;         R1 = ADDRESS OF NEXT BYTE AFTER LINE-ID
                                        ;         R4 = ADDRESS OF THE 7 WORD PARAMETER BLOCK
                                        ;
                                        ;       C-BIT SET:
                                        ;         INVALID LINE-ID SYNTAX OR UNRECOGNIZED LINE (ERROR CODE IN R0)
                                        ;
                                        ; REGISTERS:
                                        ;       R3, R4, R5 ARE PRESERVED
                                        ;-
 000000                                 PRSLIN::
 000000                                         CALL    $SAVAL                  ; SAVE ALL REGISTERS
 000004 010405                                  MOV     R4,R5                   ; COPY BUFFER POINTER
 000006 012700  000026                          MOV     #<L$PDV/2>,R0           ; SET LENGTH OF BUFFER TO CLEAR
 000012 005024                          10$:    CLR     (R4)+                   ; INITIALIZE OUTPUT AREA
 000014                                         SOB     R0,10$                  ; ...
 000020 112104                                  MOVB    (R1)+,R4                ; GET LENGTH OF STRING
 000022 010100                                  MOV     R1,R0                   ; COPY ADDRESS OF STRING
 000024 060004                                  ADD     R0,R4                   ; COMPUTE THE ADDRESS OF THE STRING END + 1
 000026 010466  000000G                         MOV     R4,R$R1(SP)             ; RETURN END OF STRING IN SAVED R1

                                                .IF DF  S$$BAS ! R$$RTR

                                                DEC     R1                      ; point to length byte
                                                CALL    $TRLIN                  ; translate logical name
                                                BCC     120$                    ; if CC, parsing all done !

                                                .ENDC   ; DF S$$BAS ! R$$RTR
                                        ;
                                        ; PARSE DEVICE NAME
                                        ;
 000032 112703  000055                          MOVB    #'-,R3                  ; SET UP ALLOWED TERMINATOR
 000036                                 20$:    CALL    CAT5                    ; CONVERT PROCESS NAME TO RAD50
 000042 103475                                  BCS     110$                    ; IF CS, ILLEGAL CHARACTER - ERROR
 000044 010165  000000                          MOV     R1,L$DDM(R5)            ; STORE RAD50 NAME
                                        ;
                                        ; PARSE CONTROLLER NUMBER
                                        ;
 000050 005702                                  TST     R2                      ; END OF THE STRING ?
 000052 001071                                  BNE     110$                    ; IF NE, YES - ERROR (NO CONTROLLER)
 000054                                         CALL    CHKMUX                  ; CHECK IF DDM IS A MUX DEVICE
 000060 132765  000004  000041                  BITB    #LP$MUX,L$PFG(R5)       ; IS IT A MUX DEVICE ?
 000066 001001                                  BNE     30$                     ; IF NE, YES - CONTINUE
```

```
PRSLIN - PARSE A LINE-ID        MACRO V05.03b Saturday 29-Jun-85 13:23  Page 13-4
Symbol table

ZF.DVP= 100000          ZF.MFL= 000010          ZF.X3P= 000000          Z.FLG   000010          Z.SCH   000007
ZF.INI= 040000          ZF.MTM= 000400          ZS.ASN= 100000          Z.LEN = 000016          $HEADR= ****** GX
ZF.KMX= 000020          ZF.MUX= 000040          ZS.BSY= 140000          Z.LLN   000006          $MUL  = ****** GX
ZF.LLC= 000004          ZF.PSE= 002000          Z.AVL   000014          Z.MAP   000020          $SAVAL= ****** GX
ZF.LMC= 000100          ZF.SLI= 010000          Z.DAT   000016          Z.NAM   000004          $$$SYS= 004374
ZF.MAN= 020000          ZF.TIM= 000200          Z.DSP   000000          Z.PCB   000012          $$$SYX= 000000


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        000772    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 217
Work  file writes: 149
Size of work file: 30099 Words  ( 118 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:32.24
SY:PRSLIN.V2,[135,134]PRSLIN/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[135,10]PRSLIN
```

```
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23     PAGE 1
```

```
                                        .SBTTL  CKEVEN - CHECK EVENT NUMBER
                                ;+
                                ;
                                ; ***  CKEVEN - CHECK EVENT NUMBER
                                ;
                                ; FUNCTION:
                                ;
                                ;   CHECK EVENT NUMBER RANGE AND SAVE IT IN EVENT MASK.
                                ;
                                ; INPUTS:
                                ;
                                ;   .PNUMB - PARSED NUMBER
                                ;
                                ; OUTPUTS:
                                ;
                                ;   AS SPECIFIED FOR LOWER LEVEL PROCEDURES
                                ;
                                ; REGISTERS R3, R4, R5 ARE PRESERVED
                                ;-
                                ;
                                ; PROCEDURE BODY
                                ;
000204                          CKEVEN::
000204  016700  000000G                 MOV     .PNUMB,R0               ;GET NUMBER
000210  004767  177564                  CALL    CKEVE                   ;PROCESS IT
000214  103002                          BCC     10$                     ;BR IF SUCCESS
000216  062716  000002                  ADD     #2,(SP)                 ;REJECT TRANSITION
000222  000207                  10$:    RETURN
```

```

                                         .SBTTL  .MOMXD - PARSE MAXIMUM DATA
                                 ;+
                                 ; ** .MOMXD - PARSE MAXIMUM DATA
                                 ;
                                 ; INPUTS        .TPARS VARIABLES.
                                 ;
                                 ; OUTPUT        $BSZ = PARSED MAXIMUM DATA
                                 ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                 ;
                                 ;-

000170                           .MOMDA::
000170  005767  000000G                  TST     .PNUMH          ; HIGH ORDER WORD
000174  001144                           BNE     ILLNUM          ; REJECT IF TOO HIGH
000176  016700  000000G                  MOV     .PNUMB,R0       ; GET PARSED VALUE
000202  020027  000025                   CMP     R0,#21.         ; CHECK LOW BOUND
000206  103537                           BLO     ILLNUM          ; IF LO, OUT OF RANGE
000210  020027  002005                   CMP     R0,#1029.       ; CHECK HIGH BOUND
000214  101134                           BHI     ILLNUM          ; IF HI, OUT OF RANGE
000216  010067  000000G                  MOV     R0,$BSZ         ; MOVE VALUE TO PARSED AREA
000222  000207                           RETURN

```

```
     57                                                 .SBTTL  DATA AREA
     58                                         ;
     59                                         ; GLOBAL DATA
     60                                         ;
     61 000000                                  .ERROR:: .BLKW  1       ; PARSE ERROR MESSAGE ADDRESS
     62 000002                                  .FLAGS:: .BLKW  1       ; POINTER TO C.STS OF CURRENT TEMPLATE FOR FLAG SETTING
     63 000004                                  .TEMP::  .BLKW  1       ; ADDRESS OF CURRENT TEMPLATE
     64                                         ;
     65                                         ; LOCAL DATA
     66                                         ;
     67 000006                                  BYTFLG: .BLKW   1       ; BYTE MODE FLAG FOR $GTNUM
```

```
    248 000402  004767  000000G                       CALL    $WDATA          ; WRITE THE RECORD TO THE OUTPUT FILE
    249 000406  103315                                BCC     10$             ; READ NEXT RECORD
    250 000410                                        ERROR$  #ERROUT         ; REPORT OUTPUT ERROR
    251 000420  000261                                SEC                     ; SET C-BIT
    252 000422  000401                                BR      80$             ; AND RETURN
    253 000424  000241                       70$:     CLC                     ; SUCCESS !!!
    254 000426  000207                       80$:     RETURN
```

Table of contents

```
    4-   57    MACRO CALLS AND DATA AREA
    5-   76    $PTNUM - PUT NUMBER INTO TEMPLATE (WORD)
    5-   77    $PTNMB - PUT NUMBER INTO TEMPLATE (BYTE)
    5-  119    $PTR50 - PUT RAD50
    6-  151    $PTOPT - CHECK FOR OPTIONAL PARAMETER
    6-  171    CHKOPT  - CHECK FOR PRECEDING OPTIONS
```

```
                                                          C 4
EVTTBL - EVENT LOGGING TABLES   MACRO V05.03b Saturday 29-Jun-85 05:16  Page 3-2
Known events table

     97 000132                                        EVENT    68,<14>
        000132  010400  040000  000000                .WORD    ^O<100>*^D<68>,$$$W0,$$$W1,$$$W2,$$$W3
        000140  000000  000000
     98 000144                                        EVENT    69,<0,16,17,18,19>
        000144  010500  000001  000017                .WORD    ^O<100>*^D<69>,$$$W0,$$$W1,$$$W2,$$$W3
        000152  000000  000000
     99 000156                                        EVENT    93,<0>
        000156  013500  000001  000000                .WORD    ^O<100>*^D<93>,$$$W0,$$$W1,$$$W2,$$$W3
        000164  000000  000000
    100 000170                                        EVENT    94,<0>
        000170  013600  000001  000000                .WORD    ^O<100>*^D<94>,$$$W0,$$$W1,$$$W2,$$$W3
        000176  000000  000000
    101 000202                                        EVENT    224,<0,1,2,3>
        000202  034000  000017  000000                .WORD    ^O<100>*^D<224>,$$$W0,$$$W1,$$$W2,$$$W3
        000210  000000  000000
    102 000214                                        EVENT    225,<0,8,9>
        000214  034100  001401  000000                .WORD    ^O<100>*^D<225>,$$$W0,$$$W1,$$$W2,$$$W3
        000222  000000  000000
    103 000226                                        EVENT    226,<0>
        000226  034200  000001  000000                .WORD    ^O<100>*^D<226>,$$$W0,$$$W1,$$$W2,$$$W3
        000234  000000  000000
    104 000240                                        EVENT    228,<0>
        000240  034400  000001  000000                .WORD    ^O<100>*^D<228>,$$$W0,$$$W1,$$$W2,$$$W3
        000246  000000  000000
    105 000252                                        EVENT    229,<0>
        000252  034500  000001  000000                .WORD    ^O<100>*^D<229>,$$$W0,$$$W1,$$$W2,$$$W3
        000260  000000  000000
    106 000264                                        EVENT    232,<0,1,2,3,4,5,6,7,8,9,10>
        000264  035000  003777  000000                .WORD    ^O<100>*^D<232>,$$$W0,$$$W1,$$$W2,$$$W3
        000272  000000  000000
    107 000276                                        EVENT    233,<0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20>
        000276  035100  177777  000037                .WORD    ^O<100>*^D<233>,$$$W0,$$$W1,$$$W2,$$$W3
        000304  000000  000000
    108
    109 000310  177777                                .WORD    -1              ; End of table
    110
    111
    112         000001                                .END
```

```
                                                          D 4
EVTTBL - EVENT LOGGING TABLES   MACRO V05.03b Saturday 29-Jun-85 05:16  Page 3-3
Symbol table
```

```
     59                                         .SBTTL  MACROS AND SYMBOLS AND INITIALIZATION
     60                                 ;
     61                                 ; DECLARE EXTERNAL MACROS
     62                                 ;
     63                                         .MCALL  ISTAT$,PRMPT$
     64                                 ;
     65                                 ; SIGNAL PROMPTING SUPPORT
     66                                 ;
     67         000001                          $GPRM   = 1
     68                                 ;
     69                                 ; MAKE TABLES READ ONLY
     70                                 ;
     71         000001                          $RONLY  = 1
```

```
                                          .SBTTL  PROMPTS
                                  ;+
                                  ;
                                  ; ***  PROMPTS
                                  ;
                                  ; FUNCTION:
                                  ;
                                  ;   DEFINE THE PROMPTS USED BY THE COMMAND PARSING TABLES.  FOR THE OVERLAID
                                  ;   NCP, THIS MUST BE ASSEMBLED TWICE.  ONCE WITH THE SYMBOL GOFF UNDEFINED
                                  ;   TO DEFINE THE PROMPT STRINGS, AND ONCE WITH THE SYMBOL GOFF DEFINED TO
                                  ;   DEFINE JUST THE PROMPT OFFSETS.
                                  ;
                                  ;   CONTAINS NO EXECUTABLE CODE.
                                  ;-

                                  ; INITIALIZE TABLE PSECTS
                                  ;
 000000                                   ISTAT$  X,Y
                                  ;
                                  ; DEFINE PROMPTS
                                  ;
 000000                                   PRMPT$  ^\''NCP>''\,                              P$NCP,  <:>

 000000                                   PRMPT$  <''Account: ''>,                           P$ACC,  <:>
 000000                                   PRMPT$  <''Active base: ''>,                       P$BSA,  <:>
 000000                                   PRMPT$  <''Active increment: ''>,                  P$INA,  <:>
 000000                                   PRMPT$  <''Active polling ratio: ''>,              P$ACT,  <:>
 000000                                   PRMPT$  <''Address: ''>,                           P$ADD,  <:>
 000000                                   PRMPT$  <''Alias name: ''>,                        P$AID,  <:>
 000000                                   PRMPT$  <''Area identification: ''>,               P$ARID, <:>
 000000                                   PRMPT$  <''Area maximum cost: ''>,                 P$AMC,  <:>
 000000                                   PRMPT$  <''Area maximum hops: ''>,                 P$AMH,  <:>
 000000                                   PRMPT$  <''Assistant option: ''>,                  P$ASTO, <:>
 000000                                   PRMPT$  <''Assistant physical address: ''>,        P$AST,  <:>
 000000                                   PRMPT$  <''Assistant node: ''>,                    P$ASTN, <:>
 000000                                   PRMPT$  <''Babble timer (sec): ''>,                P$BBT,  <:>
 000000                                   PRMPT$  <''Blocking state: ''>,                    P$BLK,  <:>
 000000                                   PRMPT$  <''Broadcast Routing Timer (sec): ''>,     P$BRT,  <:>
 000000                                   PRMPT$  <''Buffer size: ''>,                       P$BUS,  <:>
 000000                                   PRMPT$  <''Call mask: ''>,                         P$CMK,  <:>
 000000                                   PRMPT$  <''Call timer (sec): ''>,                  P$CAT,  <:>
 000000                                   PRMPT$  <''Call value: ''>,                        P$CVL,  <:>
 000000                                   PRMPT$  <''Channel number: ''>,                    P$CHN,  <:>
 000000                                   PRMPT$  <''Channel range: ''>,                     P$PCH,  <:>
 000000                                   PRMPT$  <''Circuit identification: ''>,            P$CIID, <:>
 000000                                   PRMPT$  <''Circuit owner: ''>,                     P$COWN, <:>
 000000                                   PRMPT$  <''Circuit state: ''>,                     P$CIST, <:>
 000000                                   PRMPT$  <''Circuit type: ''>,                      P$TYP,  <:>
 000000                                   PRMPT$  <''Clear circuit options: ''>,             P$CLCI, <:>
 000000                                   PRMPT$  <''Clear component: ''>,                   P$CLE,  <:>
 000000                                   PRMPT$  <''Clear line options: ''>,                P$CLLI, <:>
 000000                                   PRMPT$  <''Clear logging options: ''>,             P$CLLO, <:>
 000000                                   PRMPT$  <''Clear X25-access options: ''>,          P$CLXA, <:>
 000000                                   PRMPT$  <''Clear X25-protocol options: ''>,        P$CLXP, <:>
 000000                                   PRMPT$  <''Clear X25-server options: ''>,          P$CL5S, <:>
 000000                                   PRMPT$  <''Clear X29-server options: ''>,          P$CL9S, <:>
 000000                                   PRMPT$  <''Clear node options: ''>,                P$CLNO, <:>
```

```
                                        .SBTTL  $BL.AUT - IGNORE BLANK PROCESSING
                                        .SBTTL  $BL.A3T - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS
                                        .SBTTL  .BL.AU3 - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS AND FAIL
                                ;+
                                ; *** - $BL.AUT - IGNORE BLANK PROCESSING
                                ; *** - $BL.A3T - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS
                                ; *** - .BL.AU3 - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS AND FAIL
                                ;
                                ; INPUTS:
                                ;       NONE.
                                ;
                                ; OUTPUTS:
                                ;       .PFLAG = MODIFIED
                                ;
                                ;-
000174                          .BL.AU3::
000174  062716  000002                  ADD     #2,(SP)                 ; REJECT TRANSITION
000200                          $BL.A3T::
000200  112767  000003  000001G         MOVB    #3,.PFLAG+1             ; 3 LETTER ABBREVIATIONS
000206                          $BL.AUT::
000206  042767  000001  000000G         BIC     #1,.PFLAG               ; IGNORE BLANKS
000214  000207                          RETURN
```

```
                                                .SBTTL  .PRMWI - PARSE MAXIMUM WINDOW
                                        ;+
                                        ; ** .PRMWI - PARSE MAXIMUM WINDOW
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $WSZ = PARSED MAXIMUM WINDOW
                                        ;
                                        ;-

001370                                  .PRMWI::
001370  005767  000000G                         TST     .PNUMH          ; TEST HIGH ORDER
001374  001213                                  BNE     ILLNUM          ; IF NE, OUT OF RANGE
001376  016767  000000G 000000G                 MOV     .PNUMB,$WSZ     ; MOVE TO PARSED WINDOW SIZE
001404  001607                                  BEQ     ILLNUM          ; IF EQ, ILLEGAL VALUE
001406  026727  000000G 000177                  CMP     .PNUMB,#127.    ; CHECK HIGH BOUND
001414  101203                                  BHI     ILLNUM          ; IF HI, OUT OF RANGE
001416  000207                                  RETURN

        000001                                  .END
```

```
000232  000416                          BR      70$                     ; AND LOOK FOR TRIB #
000234  005702                  60$:    TST     R2                      ; IF NE, Wild card in bad position
000236  001054                          BNE     110$                    ; BR IF error
000240  132765  000040  000041          BITB    #LP$WCN,L$PFG(R5)       ; PREVIOUS WILD CONTROLLER # ?
000246  001050                          BNE     110$                    ; IF NE, YES - ERROR (REST MUST BE WILD)
000250                                  CALL    CDTB                    ; CONVERT LINE NUMBER
000254  103445                          BCS     110$                    ; IF CS, ILLEGAL CHARACTER - ERROR
000256  152765  000001  000041          BISB    #LP$UNT,L$PFG(R5)       ; REMEMBER UNIT # FOUND
000264  110165  000003                  MOVB    R1,L$UNT(R5)            ; STORE RESULT IN PARAMETER BLOCK

                                ;
                                ; PARSE TRIBUTARY NUMBER
                                ;
000270  005702                  70$:    TST     R2                      ; END OF THE STRING ?
000272  001030                          BNE     100$                    ; IF NE, YES - CHECK SYNTAX

                                        .IF DF R$$11S

                                        TST     ENTTYP                  ; CHECK ENTITY TYPE
                                        BMI     110$                    ; ERROR IF LINE AND NOT END

                                        .ENDC   ; DF R$$11S

000274  005003                          CLR     R3                      ; Else, CLEAR LEGAL TERMINATOR
000276                                  CALL    CHKWLD                  ; CHECK IF WILD TRIB #
000302  103404                          BCS     80$                     ; IF CS, NOT WILD
000304  152765  000202  000041          BISB    #LP$WTR!LP$TRB,L$PFG(R5); ELSE REMEMBER WILD TRIB #
000312  000416                          BR      90$                     ; AND LOOK FOR END OF STRING
000314  005702                  80$:    TST     R2                      ; If ne, wild card in bad position
000316  001024                          BNE     110$                    ; BR IF YES.
000320  132765  000140  000041          BITB    #LP$WCN!LP$WUN,L$PFG(R5) ; ANY PREVIOUS WILD FIELDS ?
000326  001020                          BNE     110$                    ; IF NE, YES-ERROR (REST MUST BE WILD)
000330                                  CALL    CDTB                    ; CONVERT TRIBUTARY ADDRESS
000334  103415                          BCS     110$                    ; IF CS, ILLEGAL CHARACTER - ERROR
000336  152765  000002  000041          BISB    #LP$TRB,L$PFG(R5)       ; REMEMBER TRIB # FOUND
000344  110165  000040                  MOVB    R1,L$TRB(R5)            ; STORE RESULT IN PARAMETER BLOCK
000350  005702                  90$:    TST     R2                      ; END OF THE STRING ?
000352  001406                          BEQ     110$                    ; IF EQ, NO - ERROR, SHOULD BE THE END

                                ;
                                ; CHECK THAT LINE EXIST, AND SYNTAX IS CORRECT
                                ;
000354                          100$:   CALL    CHKDEV                  ; ELSE, CHECK DEVICE SYNTAX
000360  103007                          BCC     120$                    ; IF NO ERROR, JUST RETURN
000362  010066  000000G         105$:   MOV     R0,R$R0(SP)             ; PLACE ERROR IN SAVED R0
000366  000404                          BR      120$                    ; AND RETURN

000370                          110$:
000370  012766  177767  000000G         MOV     #ME.IID,R$R0(SP)        ; INVALID LINE-ID IN SAVED R0
000376  000261                          SEC                             ; MARK ERROR

000400                          120$:   RETURN                          ; RETURN TO CALLER
```

```
                                                                     C 10
PRSCIR - PARSE A CIRCUIT-ID     MACRO V05.03b Saturday 29-Jun-85 13:20  Page 16-2
Symbol table

MC$W08   020000        MD.ACC   000010        ME.OPE   177747        MO$INF   000160        MP$CLN   002126
MC$W16   040000        MD.AOB   000016        ME.PLO   177751        MO$INS   000002        MP$CLT   002211
MC$W32   060000        MD.BOB   000011        ME.PMI   177743        MO$LIN   000001        MP$CMB   002142
MC$XBR   001750        MD.DIA   000007        ME.PNA   177752        MO$LOG   000001        MP$CMK   000537
MC$XBS   001751        MD.DOB   000015        ME.PRI   177775        MO$MIR   000031        MP$CMX   002153
MC$XCJ   000322        MD.DUM   000002        ME.PTY   177772        MO$NAM   000001        MP$CND   000310
MC$XCR   002260        MD.FAI   000014        ME.PVA   177760        MO$NIC   000023        MP$CNU   001753
MC$XCS   002261        MD.FNA   000001        ME.RES   177761        MO$NOD   000000        MP$COB   000311
MC$XDR   001762        MD.FOB   000006        ME.ROO   177754        MO$OFF   000001        MP$CON   002126
MC$XDS   001763        MD.LOA   000001        ME.SIZ   177774        MO$ON    000000        MP$COS   001604
MC$XFR   002272        MD.LSH   000020        ME.SYS   177746        MO$OPT   000001        MP$CPF   000230
MC$XFS   002273        MD.NOB   000012        ME.UCO   177770        MO$PER   000200        MP$CPL   000232
MC$XLJ   000323        MD.NON   177777        MF$ACT   177776        MO$PRO   000002        MP$CPT   000226
MC$XLR   002330        MD.PER   000000        MF$ADD   000000        MO$PRS   000001        MP$CPU   000161
MC$XMA   000310        MD.RES   000004        MF$ADJ   177774        MO$REA   000200        MP$CSZ   001755
MC$XMC   002305        MD.ROB   000005        MF$ALL   177775        MO$SET   000000        MP$CUS   002127
MC$XMS   002304        MD.RSH   000013        MF$BYE   000300        MO$STA   000020        MP$CVA   000540
MC$XNR   002332        MD.SDU   000005        MF$CHA   000023        MO$SUM   000000        MP$DAL   002570
MC$XRC   002316        MD.SLO   000003        MF$DUM   000020        MO$TSK   000000        MP$DCO=  001464
MC$XRR   002331        MD.SNA   000000        MF$EVT   000001        MO$VOL   000000        MP$DDT   002177
MC$XRS   002342        MD.TLO   000004        MF$FUP   000264        MO$ZER   000000        MP$DEL   001131
MC$020   000002        MD.UNA   000002        MF$KNO   177777        MP$ACB   000012        MP$DES   000156
MD$CI    000007        MD.UNR   000003        MF$LOA   000017        MP$ACC   000514        MP$DEV   002114
MD$CNA   000003        MD.UOB   000007        MF$LOO   177775        MP$ACT   004526        MP$DFA   001320
MD$COU   000001        MD.VOL   000006        MF$REA   000024        MP$ADD   000766        MP$DHO=  001465
MD$DA    000010        ME$ALI   000010        MF$SIG   177773        MP$ADJ   001440        MP$DIA   000173
MD$DL    000004        ME$AL2   000003        MF$SPF   000302        MP$ADP   006325        MP$DLB   002571
MD$DLV   000020        ME$ARE   000005        MF$SYS   000026        MP$ADS   004406        MP$DLG   004521
MD$DMC   000014        ME$CIR   000003        MF$TES   000022        MP$ALB   000036        MP$DLI=  001466
MD$DMF   000046        ME$EXA   000000        MF$TRA   000301        MP$AMC   001640        MP$DLT   002200
MD$DMP   000022        ME$EXE   000200        MF$TRI   000021        MP$AMH   001641        MP$DRO   001441
MD$DMR   000050        ME$LIN   000001        MF$ZER   000025        MP$ANB   004420        MP$DST   000454
MD$DMV   000042        ME$LOG   000002        ML$ALL   100000        MP$ASB   000024        MP$DTE   002140
MD$DN    000016        ME$MOD   000004        ML$CLS   000000        MP$ASC   004432        MP$DTY=  001452
MD$DP    000000        ME$NOD   000000        ML$CON   000001        MP$AUS   000512        MP$DUA   000207
MD$DPV   000044        ME$NON   177777        ML$EXT   000001        MP$BBT   002165        MP$DUC   000210
MD$DQ    000006        ME$OBJ   000007        ML$FIL   000002        MP$BDF   002164        MP$DUM   000202
MD$DTE   000024        ME$OB2   000004        ML$FIR   000000        MP$BFQ   002121        MP$DUP   002127
MD$DU    000002        ME$PRO   000005        ML$INT   000000        MP$BLK   001616        MP$DVC   002114
MD$DUP   000012        ME$SYS   000006        ML$KNO   140000        MP$BLO   001452        MP$DWE   001321
MD$DV    000026        ME.BLO   177744        ML$MON   000003        MP$BMX   002176        MP$ELT   000157
MD$DZ    000030        ME.CON   177753        ML$SYS   000002        MP$BNP   006330        MP$ETY   001605
MD$FUL   000000        ME.CST   177765        ML$TOP   000001        MP$BRT   001620        MP$EVE   000311
MD$HAL   000001        ME.DIS   177755        MN$UNL   000377        MP$BSA   002176        MP$FNC   001752
MD$HEL   000002        ME.DON   177600        MO$ACC   000200        MP$BSD   002203        MP$GDT   002222
MD$KCP   000013        ME.FCO   177762        MO$ADD   000002        MP$BSI   002200        MP$GNM   002223
MD$KDP   000034        ME.FIO   177756        MO$ALA   000004        MP$BSP   006331        MP$GRO   002115
MD$KDZ   000036        ME.FOP   177763        MO$ALI   000000        MP$BUF   000170        MP$GRP   000541
MD$KL    000040        ME.FOR   177776        MO$CHA   000040        MP$BUP   006324        MP$GTY   002224
MD$KMX   000054        ME.FUN   177777        MO$CIR   000003        MP$BUS   001643        MP$HAD   001757
MD$KMY   000052        ME.GRO   177745        MO$CLE   000100        MP$CAC   001750        MP$HBT   002142
MD$LEN   000050        ME.HAP   177750        MO$COU   000060        MP$CAS   001762        MP$HDD   000162
MD$PCL   000011        ME.IID   177767        MO$DAL   000000        MP$CAT   002210        MP$HTM   001612
MD$QNA   000005        ME.LCO   177766        MO$DEF   000000        MP$CCS   004406        MP$HWA   002210
MD$UNA   000001        ME.LPR   177757        MO$DPR   000000        MP$CHN   002141        MP$IAT   001322
MD$WIT   000002        ME.MPR   177773        MO$ENT   000017        MP$CIR   000144        MP$IDE   000144
MD.ABO   000017        ME.MVE   177771        MO$EVE   000100        MP$CLK   002131        MP$IDP   006327
```

```
                                                                     D 10
PRSCIR - PARSE A CIRCUIT-ID     MACRO V05.03b Saturday 29-Jun-85 13:20  Page 16-3
Symbol table
```

```
                                         .SBTTL  .EXLNK - SET MAXIMUM NUMBER OF LOGICAL LINKS
                                 ;+
                                 ; *** - .EXLNK - SET MAXIMUM NUMBER OF LOGICAL LINKS
                                 ;
                                 ; INPUT:
                                 ;       .PNUMB = LOGICAL LINKS
                                 ;
                                 ; OUTPUT:
                                 ;       $LINKS = THE MAXIMUM NUMBER OF LOGICAL LINKS
                                 ;
                                 ;-

000152                           .EXLNK::
000152  016767  000000G 000000G          MOV     .PNUMB,$LINKS   ; STORE THE LOGICAL LINK COUNT
000160  001403                           BEQ     5$              ; IF EQ, VALUE ERROR
000162  105767  000001G                  TSTB    .PNUMB+1        ; IS IT A BYTE VALUE ?
000166  001405                           BEQ     10$             ; IF EQ, YES .. OKAY
000170  062716  000002           5$:     ADD     #2,(SP)         ; ELSE REJECT TRANSITION
000174  012767  000000G 000000G          MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000202  000207                   10$:    RETURN
```

```
                                                  .SBTTL  .EXOUT - EXECUTOR OUTGOING TIMER
                                          ;+
                                          ; **    .EXOUT - EXECUTOR OUTGOING TIMER
                                          ;
                                          ; INPUTS:       TPARS CONVENTIONS.
                                          ;
                                          ; OUTPUT:       $EOUT   = PARSED EXECUTOR OUTGOING TIMER
                                          ;-
001176                                    .EXOUT::
001176  116767  000000G 000000G                   MOVB    .PNUMB,$EOUT    ; MOVE IT
001204  001406                                    BEQ     10$             ; IF EQ, ERROR
001206  005767  000000G                           TST     .PNUMH          ; TOO HIGH ?
001212  001003                                    BNE     10$             ; IF EQ, OK
001214  105767  000001G                           TSTB    .PNUMB+1        ; BYTE VALUE?
001220  001405                                    BEQ     20$             ; BR IF YES - OK

001222  012767  000000G 000000G   10$:            MOV     #ERRILN,$ERROR  ; ERROR IN NUMBER
001230  062716  000002                            ADD     #2,(SP)         ; ELSE REJECT TRANSITION

001234  000207                    20$:            RETURN
```

```
    123 000070  005003                          CLR     R3                      ; ELSE, NO - SET NEW LEGAL TERMINATOR
    124 000072                         30$:     CALL    CHKWLD                  ; IS CONTROLLER WILD ?
    125 000076  103404                          BCS     40$                     ; IF CS, NO
    126 000100  152765  000040  000041          BISB    #LP$WCN,L$PFG(R5)       ; ELSE, REMEMBER WILD CONTROLLER
    127 000106  000407                          BR      50$                     ; AND CONTINUE
    128 000110  005702                 40$:     TST     R2                      ; FATAL ERROR?
    129 000112  001051                          BNE     110$                    ; BR IF YES
    130 000114                                  CALL    CDTB                    ; CONVERT CONTROLLER NUMBER
    131 000120  103446                          BCS     110$                    ; IF CS, ERROR
    132 000122  110165  000002                  MOVB    R1,L$CTL(R5)            ; STORE RESULT IN PARAMETER BLOCK
    133 000126  132765  000004  000041 50$:     BITB    #LP$MUX,L$PFG(R5)       ; IS THIS A MUX DEVICE ?
    134 000134  001430                          BEQ     70$                     ; IF EQ, NO - LOOK FOR END OF LINE-ID
    135                                ;
    136                                ; PARSE UNIT NUMBER
    137                                ;
    138 000136  005702                          TST     R2                      ; END OF STRING ?
    139 000140  001036                          BNE     110$                    ; IF NE, YES - ERROR (NO UNIT)
    140 000142  005003                          CLR     R3                      ; SET UP ALLOWED TERMINATOR          ;01
    141 000144                                  CALL    CHKWLD                  ; CHECK IF WILD UNIT #
    142 000150  103404                          BCS     60$                     ; IF CC, NO
    143 000152  152765  000101  000041          BISB    #LP$WUN!LP$UNT,L$PFG(R5); ELSE, REMEMBER WILD UNIT NUMBER
    144 000160  000416                          BR      70$                     ; AND LOOK FOR END OF LINE-ID
    145 000162  005702                 60$:     TST     R2                      ; FATAL ERROR?
    146 000164  001024                          BNE     110$                    ; BR IF YES
    147 000166  132765  000040  000041          BITB    #LP$WCN,L$PFG(R5)       ; PREVIOUS WILD CONTROLLER # ?
    148 000174  001020                          BNE     110$                    ; IF NE, YES-ERROR (REST MUST BE WILD)
    149 000176                                  CALL    CDTB                    ; CONVERT LINE NUMBER
    150 000202  103415                          BCS     110$                    ; IF CS, ILLEGAL CHARACTER - ERROR
    151 000204  152765  000001  000041          BISB    #LP$UNT,L$PFG(R5)       ; REMEMBER UNIT # FOUND
    152 000212  110165  000003                  MOVB    R1,L$UNT(R5)            ; STORE RESULT IN PARAMETER BLOCK
    153                                ;
    154                                ; CHECK THAT LINE EXIST, AND SYNTAX IS CORRECT
    155                                ;
    156 000216  005702                 70$:     TST     R2                      ; END OF THE STRING ?
    157 000220  001406                          BEQ     110$                    ; IF EQ, NO - ERROR, SHOULD BE END
    158 000222                         100$:    CALL    CHKDEV                  ; ELSE, CHECK DEVICE SYNTAX
    159 000226  103007                          BCC     120$                    ; IF NO ERROR, JUST RETURN
    160 000230  010066  000000G                 MOV     R0,R$R0(SP)             ; PLACE ERROR IN SAVED R0
    161 000234  000404                          BR      120$                    ; AND RETURN
    162
    163 000236  012766  177767  000000G 110$:   MOV     #ME.IID,R$R0(SP)        ; INVALID LINE-ID IN SAVED R0
    164 000244  000261                          SEC                             ; MARK ERROR
    165
    166 000246                         120$:    RETURN                          ; RETURN TO CALLER
```

```
                                                                   C 14
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 1

SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE             REFERENCES

BIT0     = 000001          #6-63
BIT1     = 000002          #6-63
BIT10    = 002000          #6-63
BIT11    = 004000          #6-63
BIT12    = 010000          #6-63
BIT13    = 020000          #6-63
BIT14    = 040000          #6-63
BIT15    = 100000          #6-63
BIT2     = 000004          #6-63
BIT3     = 000010          #6-63
BIT4     = 000020          #6-63
BIT5     = 000040          #6-63
BIT6     = 000100          #6-63
BIT7     = 000200          #6-63
BIT8     = 000400          #6-63
BIT9     = 001000          #6-63
B$10       000003          #5-63
B$11       000004          #5-63
B$12       000005          #5-63
B$5        000000          #5-63
B$6        000001          #5-63
B$7        000002          #5-63
B$7S10     000006          #5-63
CAT5       000256 R         7-112       #8-199
CDTB       000400 R         7-130        7-149       #9-264
CHKDEV     000530 R         7-158       #11-340
CHKMUX     000476 R         7-120       #10-309
CHKWLD     000730 R         7-124        7-141       #13-448
CVTBL      000250 R        #8-195        8-202        8-220
I$$AS    = ******           5-59
LF$END   = 000020          #6-63
LF$MLT   = 040000          #6-63
LF$REA   = 000001          #6-63
LF$SEG   = 100000          #6-63
LF$SIG   = 000040          #6-63
LF$SKP   = 000004          #6-63
LF$VR2   = 000010          #6-63
LF$ZER   = 000002          #6-63
LF.ACT   = 100000          #5-60
LF.BRO   = 000400          #5-60
LF.BWT   = 000007          #5-60
LF.ENA   = 002000          #5-60
LF.LPB   = 001000          #5-60
LF.MDC   = 000100          #5-60
LF.MFL   = 004000          #5-60
LF.MTP   = 000020          #5-60
LF.PAC   = 000200          #5-60
LF.RDY   = 040000          #5-60
LF.REA   = 010000          #5-60
LF.SER   = 000040          #5-60
LF.TIM   = 000010          #5-60
LF.UNL   = 020000          #5-60
```

```
                                                                   D 14
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 2
```

```
                                        .SBTTL  CKEVER - CHECK EVENT RANGE
                                ;+
                                ;
                                ; ***  CKEVER - CHECK EVENT RANGE
                                ;
                                ; FUNCTION:
                                ;
                                ;   CHECK VALIDITY OF AN EVENT RANGE AND SET BITS IN EVENT MASK.
                                ;
                                ; INPUTS:
                                ;
                                ;   $EVBEG - 1ST NUMBER IN RANGE
                                ;   .PNUMB - LOW ORDER WORD OF 2ND NUMBER IN RANGE
                                ;   .PNUMH - HIGH ORDER WORD OF 2ND NUMBER IN RANGE
                                ;
                                ; OUTPUTS:
                                ;
                                ;   AS SPECIFIED FOR LOWER LEVEL PROCEDURES
                                ;
                                ; REGISTERS R3, R4, R5 ARE PRESERVED
                                ;-
                                ;
                                ; PROCEDURE BODY
                                ;
000224                          CKEVER::
000224  016700  000000G                 MOV     $EVBEG,R0               ;GET 1ST NUMBER
000230  005767  000000G                 TST     .PNUMH                  ;CHECK HIGH ORDER WORD
000234  001012                          BNE     20$                     ;BR IF HIGH ORDER WORD <> 0
000236  026700  000000G                 CMP     .PNUMB,R0               ;COMPARE 2ND NUMBER TO 1ST NUMBER
000242  003407                          BLE     20$                     ;BR IF 1ST NUMBER > 2ND NUMBER
000244  020067  000000G         10$:    CMP     R0,.PNUMB               ;COMPARE CURRENT NUMBER TO 2ND NUMBER
000250  003006                          BGT     30$                     ;FALSE IF CURRENT NUMBER > 2ND NUMBER
000252  004767  177522                  CALL    CKEVE                   ;PROCESS NUMBER
000256  005200                          INC     R0                      ;INCREMENT CURRENT NUMBER
000260  103371                          BCC     10$                     ;BR IF MORE TO PROCESS
000262  062716  000002          20$:    ADD     #2,(SP)         ;REJECT TRANSITION
000266  000207                  30$:    RETURN
```

```
                                          .SBTTL  .MODFW - PARSE DEFAULT WINDOW
                                  ;+
                                  ; ** .MODFW - PARSE DEFAULT WINDOW
                                  ;
                                  ; INPUTS        .TPARS VARIABLES.
                                  ;
                                  ; OUTPUT        $DFW = PARSED MAXIMUM DATA
                                  ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                  ;-
 000224                           .MODFW::
 000224  005767  000000G                  TST     .PNUMH          ; HIGH ORDER WORD
 000230  001126                           BNE     ILLNUM          ; REJECT IF TOO HIGH
 000232  026727  000000G 000177           CMP     .PNUMB,#127.    ; CHECK HIGH BOUND
 000240  003122                           BGT     ILLNUM          ; BR IF  ILLEGAL NUMBER
 000242  116767  000000G 000000G          MOVB    .PNUMB,$DFW     ; MOVE VALUE TO PARSED AREA
 000250  001516                           BEQ     ILLNUM          ; BR IF ZERO
 000252  000207                           RETURN
```

```
                                        .SBTTL  $GTNUM - GET NUMBER INTO TEMPLATE (WORD)
                                        .SBTTL  $GTNMB - GET NUMBER INTO TEMPLATE (BYTE)
                                ;+
                                ; ** $GTNUM - GET NUMBER INTO TEMPLATE (WORD)
                                ; ** $GTNMB - GET NUMBER INTO TEMPLATE (BYTE)
                                ;
                                ; INPUTS:       .TEMP = TEMPLATE ADDRESS
                                ;               R2 = OFFSET IN TEMPLATE
                                ;               R1 = HIGH LIMIT FOR RANGE CHECK
                                ;               R0 = LOW LIMIT FOR RANGE CHECK.
                                ;               .TPARS VARIABLES.
                                ;
                                ; OUTPUT:       IF VALUE IN RANGE IT IS LOADED INTO THE TEMPLATE,
                                ;               OTHERWISE THE TRANSITION IS REJECTED.
                                ;
                                ; REGISTERS:    R2 = ADDRESS OF TARGET FIELD
                                ;
                                ; CALLING SEQUENCE:  ENTER VIA CALLR FROM WITHIN A ACTION ROUTINE.
                                ;
                                ;-
                                        .ENABL  LSB

 000010                         $GTNMB::
 000010  010667  177772                 MOV     SP,BYTFLG       ; SET UP BYTE MODE
 000014  000402                         BR      10$             ; JOIN COMMON CODE

 000016                         $GTNUM::
 000016  005067  177764                 CLR     BYTFLG          ; SET UP WORD MODE
 000022  005767  000000G        10$:    TST     .PNUMH          ; 32 BIT VALUE ?
 000026  001064                         BNE     REJECT          ; IF NE, YES - REJECT TRANSITION
 000030  026700  000000G                CMP     .PNUMB,R0       ; CHECK LOW BOUND
 000034  103461                         BLO     REJECT          ; IF LO, OUT OF RANGE
 000036  026701  000000G                CMP     .PNUMB,R1       ; CHECK HIGH BOUND
 000042  101056                         BHI     REJECT          ; IF HI, OUT OF RANGE
 000044  066702  177734                 ADD     .TEMP,R2        ; POINT TO TARGET FIELD
 000050  005767  177732                 TST     BYTFLG          ; IS IT BYTE MODE ?
 000054  001003                         BNE     20$             ; IF NE, YES
 000056  016712  000000G                MOV     .PNUMB,(R2)     ; STUFF WORD INTO TEMPLATE
 000062  000402                         BR      30$             ; EXIT
 000064  116712  000000G        20$:    MOVB    .PNUMB,(R2)     ; STUFF BYTE INTO TEMPLATE
 000070  000207                 30$:    RETURN

                                        .DSABL  LSB

                                        .SBTTL  $GTR50 - GET RAD50
                                ;+
                                ; ** $GTR50 - GET RAD50
                                ;
                                ; INPUTS:       .TEMP = TEMPLATE ADDRESS
                                ;               R2 = OFFSET IN TEMPLATE
                                ;               R1 = SIZE OF FIELD IN WORDS.
                                ;               .TPARS VARIABLES.
                                ;
                                ; OUTPUT:       IF THE LENGTH OF THE MATCHED STRING EXCEEDS THE
                                ;               FIELD SIZE THE TRANSITION IS REJECTED,
                                ;               OTHERWISE FIELD IS LOADED WITH THE MATCHED STRING IN RAD50
                                ;               FORMAT.
```

```
                                                    .SBTTL  SKIPIT - DON'T WRITE OUT THE LINE (ZERO $LEN)
                                            ;+
                                            ; *** - SKIPIT - DON'T WRITE OUT THE LINE
                                            ;
                                            ; INPUT:
                                            ;       $LEN = THE LENGTH OF THE LINE READ IN
                                            ;
                                            ; OUTPUT:
                                            ;       $LEN = ZERO (FORCES THE LINE NOT TO BE WRITTEN OUT
                                            ;
                                            ;-

    000430                                  SKIPIT:
    000430  005067  177346                          CLR     $LEN            ; DON'T WANT TO WRITE OUT THIS LINE
    000434  000207                                  RETURN
```

```
          .TITLE  CFEUSB - TEMPLATE UPDATE SUBROUTINES
          .IDENT  /V05.00/
          .ENABL  LC

;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       CETAB TEMPLATE TEMPLATE UPDATE SUBROUTINES
;
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  1 -AUG-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                        D  4
EVTTBL - EVENT LOGGING TABLES   MACRO V05.03b Saturday 29-Jun-85 05:16  Page 3-3
Symbol table

$KNEVT   000000RG        $$$B12= 010000         $$$B3 = 000010         $$$B7 = 000200         $$$W1 = 000037
$$$B0 = 000001           $$$B13= 020000         $$$B4 = 000020         $$$B8 = 000400         $$$W2 = 000000
$$$B1 = 000002           $$$B14= 040000         $$$B5 = 000040         $$$B9 = 001000         $$$W3 = 000000
$$$B10= 002000           $$$B15= 100000         $$$B6 = 000100         $$$W0 = 177777         ...N  = 000000
$$$B11= 004000           $$$B2 = 000004

. ABS.  000000     000   (RW,I,GBL,ABS,OVR)
        000312     001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8330 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.44
SY:EVTTBL.V2,[132,134]EVTTBL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]EVTTBL
```

```
                                                        E  4
EVTTBL      CREATED BY  MACRO  ON 29-JUN-85 AT 05:16      PAGE 1
```

```
                                  .SBTTL  PROMPTS
                           ;+
                           ;
                           ; ***  PROMPTS
                           ;
                           ; FUNCTION:
                           ;
                           ;   DEFINE THE PROMPTS USED BY THE COMMAND PARSING TABLES.  FOR THE OVERLAID
                           ;   NCP, THIS MUST BE ASSEMBLED TWICE.  ONCE WITH THE SYMBOL GOFF UNDEFINED
                           ;   TO DEFINE THE PROMPT STRINGS, AND ONCE WITH THE SYMBOL GOFF DEFINED TO
                           ;   DEFINE JUST THE PROMPT OFFSETS.
                           ;
                           ;   CONTAINS NO EXECUTABLE CODE.
                           ;-
                           ;
                           ; INITIALIZE TABLE PSECTS
                           ;
000000                            ISTAT$  X,Y
                           ;
                           ; DEFINE PROMPTS
                           ;
000000                            PRMPT$  ^\'NCP>'\,                              P$NCP,  <:>

000000                            PRMPT$  <'Account: '>,                          P$ACC,  <:>
000000                            PRMPT$  <'Active base: '>,                      P$BSA,  <:>
000000                            PRMPT$  <'Active increment: '>,                 P$INA,  <:>
000000                            PRMPT$  <'Active polling ratio: '>,             P$ACT,  <:>
000000                            PRMPT$  <'Address: '>,                          P$ADD,  <:>
000000                            PRMPT$  <'Alias name: '>,                       P$AID,  <:>
000000                            PRMPT$  <'Area identification: '>,              P$ARID, <:>
000000                            PRMPT$  <'Area maximum cost: '>,                P$AMC,  <:>
000000                            PRMPT$  <'Area maximum hops: '>,                P$AMH,  <:>
000000                            PRMPT$  <'Assistant option: '>,                 P$ASTO, <:>
000000                            PRMPT$  <'Assistant physical address: '>,      P$AST,  <:>
000000                            PRMPT$  <'Assistant node: '>,                   P$ASTN, <:>
000000                            PRMPT$  <'Babble timer (sec): '>,               P$BBT,  <:>
000000                            PRMPT$  <'Blocking state: '>,                   P$BLK,  <:>
000000                            PRMPT$  <'Broadcast Routing Timer (sec): '>,    P$BRT,  <:>
000000                            PRMPT$  <'Buffer size: '>,                      P$BUS,  <:>
000000                            PRMPT$  <'Call mask: '>,                        P$CMK,  <:>
000000                            PRMPT$  <'Call timer (sec): '>,                 P$CAT,  <:>
000000                            PRMPT$  <'Call value: '>,                       P$CVL,  <:>
000000                            PRMPT$  <'Channel number: '>,                   P$CHN,  <:>
000000                            PRMPT$  <'Channel range: '>,                    P$PCH,  <:>
000000                            PRMPT$  <'Circuit identification: '>,           P$CIID, <:>
000000                            PRMPT$  <'Circuit owner: '>,                    P$COWN, <:>
000000                            PRMPT$  <'Circuit state: '>,                    P$CIST, <:>
000000                            PRMPT$  <'Circuit type: '>,                     P$TYP,  <:>
000000                            PRMPT$  <'Clear circuit options: '>,            P$CLCI, <:>
000000                            PRMPT$  <'Clear component: '>,                  P$CLE,  <:>
000000                            PRMPT$  <'Clear line options: '>,               P$CLLI, <:>
000000                            PRMPT$  <'Clear logging options: '>,            P$CLLO, <:>
000000                            PRMPT$  <'Clear X25-access options: '>,         P$CLXA, <:>
000000                            PRMPT$  <'Clear X25-protocol options: '>,       P$CLXP, <:>
000000                            PRMPT$  <'Clear X25-server options: '>,         P$CL5S, <:>
000000                            PRMPT$  <'Clear X29-server options: '>,         P$CL9S, <:>
000000                            PRMPT$  <'Clear node options: '>,               P$CLNO, <:>
```

```
    130 000000                          PRMPT$  <"Clear object options: ">,             P$CLOB, <:>
    131 000000                          PRMPT$  <"Clear timer (sec): ">,                P$CLT,  <:>
    132 000000                          PRMPT$  <"Clear trace options: ">,              P$CLTR, <:>
    133 000000                          PRMPT$  <"Clock mode: ">,                       P$CLK,  <:>
    134 000000                          PRMPT$  <"Command: ">,                          P$COM,  <:>
    135 000000                          PRMPT$  <"Component type: ">,                   P$PEN,  <:>
    136 000000                          PRMPT$  <"Controller mode: ">,                  P$CON,  <:>
    137 000000                          PRMPT$  <"Copies: ">,                           P$COP,  <:>
    138 000000                          PRMPT$  <"Cost: ">,                             P$COS,  <:>
    139 000000                          PRMPT$  <"Counter timer (sec): ">,              P$CTM,  <:>
    140 000000                          PRMPT$  <"CPU type: ">,                         P$CPU,  <:>
    141 000000                          PRMPT$  <"CSR address: ">,                      P$CSR,  <:>
    142 000000                          PRMPT$  <"Dead polling ratio: ">,               P$DEA,  <:>
    143 000000                          PRMPT$  <"Dead threshold: ">,                   P$DDTH, <:>
    144 000000                          PRMPT$  <"Dead timer (msec): ">,                P$DDT,  <:>
    145 000000                          PRMPT$  <"Default data (bytes): ">,             P$BDF,  <:>
    146 000000                          PRMPT$  <"Default window (bytes): ">,           P$WDF,  <:>
    147 000000                          PRMPT$  <"Delay factor: ">,                     P$DFA,  <:>
    148 000000                          PRMPT$  <"Delay timer (msec): ">,               P$DLT,  <:>
    149 000000                          PRMPT$  <"Delay weight: ">,                     P$DWE,  <:>
    150 000000                          PRMPT$  <"Destination id:">,                    P$DST,  <:>
    151 000000                          PRMPT$  <"Destination node: ">,                 P$DES,  <:>
    152 000000                          PRMPT$  <"Device spec: ">,                      P$DEV,  <:>
    153 000000                          PRMPT$  <"Diagnostic file: ">,                  P$DIA,  <:>
    154 000000                          PRMPT$  <"DTE address: ">,                      P$DTE,  <:>
    155 000000                          PRMPT$  <"DTE number: ">,                       P$NUM,  <:>
    156 000000                          PRMPT$  <"Dump address: ">,                     P$DUA,  <:>
    157 000000                          PRMPT$  <"Dump count: ">,                       P$DUC,  <:>
    158 000000                          PRMPT$  <"Dump file: ">,                        P$DUM,  <:>
    159 000000                          PRMPT$  <"Dump option: ">,                      P$DUMO, <:>
    160 000000                          PRMPT$  <"Duplex: ">,                           P$DUP,  <:>
    161 000000                          PRMPT$  <"Dying base: ">,                       P$BSD,  <:>
    162 000000                          PRMPT$  <"Dying increment: ">,                  P$IND,  <:>
    163 000000                          PRMPT$  <"Dying threshold: ">,                  P$DYTH, <:>
    164 000000                          PRMPT$  <"Event identification: ">,             P$EVE,  <:>
    165 000000                          PRMPT$  <"File name: ">,                        P$FNAM, <:>
    166 000000                          PRMPT$  <"Group name: ">,                       P$GRP,  <:>
    167 000000                          PRMPT$  <"Group number: ">,                     P$GNM,  <:>
    168 000000                          PRMPT$  <"Hello timer (sec): ">,                P$HTM,  <:>
    169 000000                          PRMPT$  <"Hardware address: ">,                 P$HAD,  <:>
    170 000000                          PRMPT$  <"Hardware option: ">,                  P$HAO,  <:>
    171 000000                          PRMPT$  <"Hexadecimal password: ">,             P$HPAS, <:>
    172 000000                          PRMPT$  <"Holdback timer (msec): ">,            P$HBT,  <:>
    173 000000                          PRMPT$  <"Host node identification: ">,         P$HOS,  <:>
    174 000000                          PRMPT$      entification: ">,                   P$IDE,  <:>
    175 000000                          PRMPT$      active base: ">,                    P$BSI,  <:>
    176 000000                          PRMPT$      active increment: ">,               P$INI,  <:>
    177 000000                          PRMPT$      active threshold: ">,               P$INTH, <:>
    178 000000                          PRMPT$  <"Inactivity timer (sec): ">,           P$IAT,  <:>
    179 000000                          PRMPT$  <"Incoming timer (sec): ">,             P$ITI,  <:>
    180 000000                          PRMPT$  <"Length of test data: ">,              P$LEN,  <:>
    181 000000                          PRMPT$  <"Line identification: ">,              P$LNID, <:>
    182 000000                          PRMPT$  <"Line owner: ">,                       P$LOWN, <:>
    183 000000                          PRMPT$  <"Line protocol: ">,                    P$PRO,  <:>
    184 000000                          PRMPT$  <"Line state: ">,                       P$LIST, <:>
    185 000000                          PRMPT$  <"Load image input file: ">,            P$LOA,  <:>
    186 000000                          PRMPT$  <"Load options: ">,                     P$LDOP, <:>
```

```
                                        .SBTTL  .HLPAC - HELP ACTION ROUTINE
                                ;+
                                ; *** - .HLPAC - HELP ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       R4 = ADDRESS OF LINE AFTER "HELP"
                                ;
                                ; OUTPUTS:
                                ;       $CMDPT = ADDRESS OF LINE AFTER "HELP"
                                ;
                                ;-

 000216                         .HLPAC::
 000216  010467  000000G                MOV     R4,$CMDPT
 000222  000207                         RETURN
```

```
PRSACT - PARSE ACTION ROUTINES  MACRO V05.03b Saturday 29-Jun-85 05:17  Page 18-1
Symbol table

CM.DEF= ****** GX       ERRNAM  000000R         $BSZ  = ****** GX       $TMLST= ****** GX       .PRMDA  001322RG
CM.LIS= ****** GX       ERRNOD= ****** GX       $CAT5 = ****** GX       $WSZ  = ****** GX       .PRMWI  001370RG
CO.EXE= ****** GX       EX$ERR= 000002          $CMAND= ****** GX       .BL.AU  000174RG        .PSTCN= ****** GX
CS.REM= ****** GX       FNDNAM  001254R         $CMDPT= ****** GX       .EXADD  000764RG        .PSTPT= ****** GX
C.ADD   000010          ILLNUM  001024R         $COMP = ****** GX       .EXNAM  000344RG        .RNADD  000612RG
C.NAM   000004          NODNAM  000374R         $ERROR= ****** GX       .EXRTM  001040RG        .RNAD2  000460RG
C.STS = ****** GX       START = ****** GX       $NODAD= ****** GX       .HLPAC  000216RG        .RNAD3  000650RG
ERRADD  000041R         $BL.AU  000206RG        $NODNA= ****** GX       .NONAM  000224RG        .RNHNA  001172RG
ERREXE  000105R         $BL.A3  000200RG        $NTADD= ****** GX       .PFLAG= ****** GX       .RNINI  000452RG
ERRILC= ****** GX       $BL.MA  000156RG        $NTNAM= ****** GX       .PNUMB= ****** GX       .RNNAM  001052RG
ERRILN= ****** GX       $BL.MN  000150RG        $RTM  = ****** GX       .PNUMH= ****** GX

. ABS.  000012    000   (RW,I,GBL,ABS,OVR)
        001420    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 211 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:12.17
SY:PRSACT.V2,[132,134]PRSACT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSACT
```

```
PRSACT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:17     PAGE 1
```

```
                                         .IF NDF R$$11S
                                         .SBTTL  STOCIR - Store circuit-id in context area
                               ;+
                               ; *** STOCIR - Store circuit-id in context area
                               ;
                               ; INPUTS
                               ;         R1 - Pointer to counted id-string
                               ;         R5 - Address of context area
                               ;
                               ; OUTPUTS
                               ;         String copied to L$NAM(R5) and padded with spaces
                               ;         Registers preserved
                               ;-

000402                         STOCIR::
000402                                   SAVRG   <R0,R1,R2>

000410  010502                           MOV     R5,R2                   ; Copy context area pointer
000412  062702  000000                   ADD     #L$NAM,R2               ; Point to circuit name field
000416  012700  000020                   MOV     #L$NLEN,R0              ; Max length of name
000422  112722  000040         10$:      MOVB    #SPACE,(R2)+
000426                                   SOB     R0,10$

000432  010502                           MOV     R5,R2
000434  062702  000000                   ADD     #L$NAM,R2
000440  112100                           MOVB    (R1)+,R0                ; Get actual length of string
000442  112122                 20$:      MOVB    (R1)+,(R2)+             ; Copy circuit name
000444                                   SOB     R0,20$

000450                                   RESRG   <R2,R1,R0>
000456                                   RETURN
```

```
MP$IHO    000215        MP$NAP    006326        MP$SCA    000310        MS$ATR    000011        MT$ROU    000000
MP$INA    002177        MP$NCT    000240        MP$SCO    000144        MS$AUT    000000        MT$R04    000004
MP$IND    002204        MP$NET    002114        MP$SCT    002176        MS$CLE    000003        MT$SEC    000000
MP$INI    002201        MP$NLI    000765        MP$SDU    000203        MS$DED    000004        MT$SGD    000020
MP$ITI    000776        MP$NNA    000764        MP$SDV    000160        MS$DIE    000003        MT$SYS    000002
MP$LAA    000231        MP$NOD    000500        MP$SEH    004401        MS$DUM    000004        MT$TER    000001
MP$LAN    000234        MP$NRA    006334        MP$SER    000144        MS$FAI    000013        MT$TRI    000002
MP$LAR    000202        MP$NRB    000214        MP$SET    000000        MS$GLO    000000        MT$TYP    007777
MP$LAS    000012        MP$NSA    001617        MP$SGZ    001644        MS$HOL    000002        MT$USD    000000
MP$LBS    000203        MP$NTI    002141        MP$SID    000176        MS$INA    000002        MT$X25    000003
MP$LCO    000226        MP$NUM    001642        MP$SIN    000310        MS$LOA    000003        MU$INC    000001
MP$LCT    000156        MP$NVE    001274        MP$SLI    000156        MS$LOO    000002        MU$OUT    000002
MP$LHL    00023         MP$NXN=   001476        MP$SLO    000171        MS$OFF    000001        MU$PER    000000
MP$LLE    000227        MP$OAC    000632        MP$SMX    002202        MS$ON     000000        MV$III    000000
MP$LLO    004432        MP$OAN    000620        MP$SND    000500        MS$PRO    000006        MV$IV     000001
MP$LMB    002152        MP$OCO    000776        MP$SNP    006332        MS$REA    000004        MX$ACT    000020
MP$LMX    002200        MP$OHO    000214        MP$SNU    000542        MS$REF    000001        MX$CIR    000020
MP$LNA    000144        MP$ONA    000764        MP$SOB    000524        MS$RES    000003        MX$CLN    000020
MP$LNO    000233        MP$ONR    002114        MP$SPA    000157        MS$RST    000001        MX$CNM    000006
MP$LOA    000170        MP$OQL    001453        MP$SPR    000536        MS$RSX    000002        MX$CON    000006
MP$LOG    004514        MP$OTI    000777        MP$SPS    000513        MS$RT     000005        MX$DAC    000020
MP$LOD    000620        MP$OTY    001022        MP$STA    000000        MS$SER    000002        MX$DTE    000020
MP$LTM    001613        MP$OUS    000777        MP$STI    002140        MS$SHU    000002        MX$FIL    000034
MP$LTY    004533        MP$OVE    001010        MP$STT    002201        MS$SRV    000007        MX$LGC    000777
MP$LWI    000230        MP$OWN    004374        MP$STY    000175        MS$STA    000000        MX$LIN    000020
MP$MAC    004420        MP$PAR    000036        MP$SUB    000001        MS$SYN    000012        MX$LON    000006
MP$MAD    001630        MP$PAS    000513        MP$SUR    000156        MS$TER    000001        MX$NMS    000454
MP$MAP    006323        MP$PCH    002152        MP$SUS    000512        MS$TOP    000003        MX$NOD    000006
MP$MAR    001635        MP$PCO    000024        MP$SVR    000163        MS$TRI    000005        MX$OBJ    000006
MP$MAV    006321        MP$PCT    000144        MP$SWI    002260        MS$UNR    000005        MX$OWN    000040
MP$MBN    001636        MP$PDT    002114        MP$SYP    002424        MS$VMS    000004        MX$PAR    000036
MP$MBR    001637        MP$PHA    000170        MP$TFL    000156        MS.DON    177600        MX$PAS    000010
MP$MBU    001642        MP$PHY    000012        MP$TH1    002206        MS.MCR    000002        MX$RAC    000047
MP$MCB    000156        MP$PLI    000025        MP$TH2    002205        MS.PAR    000003        MX$RID    000047
MP$MCO    001632        MP$PLN    002140        MP$TH3    002202        MS.SUC    000001        MX$RPS    000047
MP$MDE    004421        MP$PLO    000012        MP$TLN    000202        MT$AR4    000003        MX$SID    000040
MP$MHO    001633        MP$PMC    004374        MP$TLO    000172        MT$ASC    000100        MX$SNK    000377
MP$MLB    000202        MP$PNT    002126        MP$TPA    004375        MT$BIL    000001        MX$TYP    000077
MP$MLK    001306        MP$PRI    004411        MP$TRI    002164        MT$BYS    000011        MX$UID    000020
MP$MLN    001631        MP$PRO    002130        MP$TST    000144        MT$CI     000007        M$$CRB=   000124
MP$MLP    006333        MP$PSS    001763        MP$TYP    002130        MT$CLE    000077        M$$CRX=   000000
MP$MRB    002171        MP$PST    001762        MP$UCS    004407        MT$COD    000200        M$$FCS=   000000
MP$MRO    001605        MP$RET    001631        MP$USR    001750        MT$CON    000001        M$$MGE=   000000
MP$MRP    006322        MP$RFA    001323        MP$VEC    004410        MT$DMC    000004        M$$NET=   000000
MP$MRT    002153        MP$RMX    002201        MP$VER    004406        MT$ETH    000006        M$$OVR=   000000
MP$MRV    006320        MP$RPA    004374        MP$WDF    002165        MT$HEX    000040        NEWPVC    000460R
MP$MSB    000170        MP$RPR    001606        MP$WMX    002177        MT$LPB    000005        NM.CIN    003400
MP$MVE    000145        MP$RRT    002213        MP$XMT    002166        MT$MAX    000037        NM.INI    002400
MP$MVI    001634        MP$RST    002212        MP$XPF    000242        MT$MUL    000100        NM.OPR    003000
MP$MV     001751        MP$RSV    001754        MP$XPL    000244        MT$NLE    000017        NM.VR2    002401
MP$ML     002154        MP$RSZ    001756        MP$XPT    000240        MT$NON    000001        NM.VR3    002402
MP$M      002172        MP$RTI    001616        MP$XXX    177777        MT$NR4    000005        N$$ACC=   000001
MP$M      000466        MP$RTM    001611        MP$5ST    004406        MT$NTY    000060        N$$BUF=   000001
MP$M      001630        MP$RVE    001604        MS$ACT    000001        MT$OCT    000060        N$$LDV=   000001
MP$M      002143        MP$RVT    000156        MS$ADU    000010        MT$PHA    000002        N$$MCP=   000001
MP$N      006335        MP$SAC    000514        MS$ALO    000007        MT$POI    000000        N$$MLL=   000001
MP$N      001130        MP$SAD    000543        MS$ASE    000006        MT$QP2    000010        N$$MOV=   000010
```

```
    146                                                 .SBTTL  .EXHOP - SET MAXIMUM NUMBER OF HOPS IN NETWORK
    147                                         ;+
    148                                         ; *** - .EXHOP - SET MAXIMUM NUMBER OF HOPS IN NETWORK
    149                                         ;
    150                                         ; INPUTS:
    151                                         ;       .PNUMB = THE MAXIMUM NUMBER OF HOPS IN NETWORK
    152                                         ;
    153                                         ; OUTPUTS:
    154                                         ;       $MAXH = THE MAXIMUM NUMBER OF HOPS IN NETWORK
    155                                         ;       THE  TRANSITION IS ACCEPTED/REJECTED.
    156                                         ;
    157                                         ;-
    158
    159 000204                                  .EXHOP::
    160 000204  016767  000000G 000000G                 MOV     .PNUMB,$MAXH    ; STORE THE MAX. NUMBER OF HOPS IN NETWRK
    161 000212  001404                                  BEQ     10$             ; IF EQ, ILLEGAL VALUE
    162 000214  026727  000000G 000000G                 CMP     .PNUMB,#MAXHOP  ; IS THE VALUE IN RANGE ?
    163 000222  101405                                  BLOS    20$             ; IF LOS, YES .. OKAY
    164 000224  062716  000002                  10$:    ADD     #2,(SP)         ; ELSE, REJECT TRANSITION
    165 000230  012767  000000G 000000G                 MOV     #ERRILN,$ERROR  ; INVALID NUMBER
    166 000236  000207                          20$:    RETURN
    496                                                 .SBTTL  .EXRET - EXECUTOR RETRANSMIT FACTOR
    497                                         ;+
    498                                         ; **    .EXRET - EXECUTOR RETRANSMIT FACTOR
    499                                         ;
    500                                         ; INPUTS:       TPARS CONVENTIONS.
    501                                         ;
    502                                         ; OUTPUT:       $ERET   = PARSED EXECUTOR RETRANSMIT FACTOR
    503                                         ;-
    504 001236                                  .EXRET::
    505 001236  116767  000000G 000000G                 MOVB    .PNUMB,$ERET    ; MOVE IT
    506 001244  001406                                  BEQ     10$             ; IF EQ, ERROR
    507 001246  005767  000000G                         TST     .PNUMH          ; TOO HIGH ?
    508 001252  001003                                  BNE     10$             ; IF EQ, OK
    509 001254  105767  000001G                         TSTB    .PNUMB+1        ; BYTE VALUE?
    510 001260  001405                                  BEQ     20$             ; BR IF YES - OK
    511
    512 001262  012767  000000G 000000G 10$:            MOV     #ERRILN,$ERROR  ; ERROR IN NUMBER
    513 001270  062716  000002                          ADD     #2,(SP)         ; ELSE REJECT TRANSITION
    514
    515 001274  000207                  20$:            RETURN
    516                 000001                          .END
```

```
                                 .SBTTL  CAT5 - CONVERT ASCII TO RAD50
                         ;+
                         ; **-CAT5-CONVERT ASCII TO RAD50
                         ;
                         ; INPUTS:
                         ;
                         ;        R0 = ADDRESS OF FIRST BYTE IN STRING
                         ;        R3 = LEGAL TERMINATING CHARACTER
                         ;        R4 = ADDRESS OF THE END OF THE STRING + 1
                         ;
                         ;
                         ; OUTPUTS:
                         ;
                         ;        C-BIT CLEAR:
                         ;        R0 = ADDRESS OF NEXT BYTE IN STRING
                         ;        R1 = CONVERTED RESULT
                         ;        R2 = COMPLETION CODE (0 = MORE IN THE STRING)
                         ;                             (NOT 0 = NO MORE IN THE STRING)
                         ;
                         ;        C-BIT SET:
                         ;        ILLEGAL CHARACTER IN THE STRING
                         ;
                         ; REGISTERS MODIFIED:
                         ;
                         ;        NONE
                         ;
                         ;-

000250  003100           CVTBL:  .WORD   50*50
000252  000050                   .WORD   50
000254  000001                   .WORD   1

000256  010546           CAT5:   MOV     R5,-(SP)        ; SAVE REGISTERS
000260  005046                   CLR     -(SP)           ; ESTABLISH EXIT CONDITION FLAG
000262  005001                   CLR     R1              ; CLEAR ACCUMULATION
000264  012702  000250'          MOV     #CVTBL,R2       ; SET ADDRESS OF SCALING TABLE
000270  112005           10$:    MOVB    (R0)+,R5        ; GET A CHARACTER
000272  020400                   CMP     R4,R0           ; REACHED THE END OF THE STRING ?
000274  103434                   BLO     30$             ; IF LO, YES - END OF THE STRING
000276  122705  000132           CMPB    #'Z,R5          ; IS IT ALPHABETIC ? (BELOW Z)
000302  103425                   BLO     20$             ; IF LO, NO - ERROR
000304  122705  000101           CMPB    #'A,R5          ; IS IT BELOW A ?
000310  101022                   BHI     20$             ; IF HI, YES - ERROR
000312  162705  000100           SUB     #100,R5         ; SCALE TO RAD50
000316  010046                   MOV     R0,-(SP)        ; SAVE R0
000320  012200                   MOV     (R2)+,R0        ; GET SCALE FACTOR
000322  010146                   MOV     R1,-(SP)        ; SAVE R1
000324  010501                   MOV     R5,R1           ; COPY CHARACTER
000326                           CALL    $MUL            ; MULTIPLY IT OUT
000332  010105                   MOV     R1,R5           ; COPY RESULT
000334  012601                   MOV     (SP)+,R1        ; RESTORE ACCUMULATION
000336  012600                   MOV     (SP)+,R0        ; RESTORE STRING POINTER
000340  060501                   ADD     R5,R1           ; ACCUMULATE THE RESULT
000342  020227  000256'          CMP     R2,#CVTBL+6     ; CONVERTED ENOUGH ?
000346  103750                   BLO     10$             ; IF LO, NO - KEEP GOING
000350  112005                   MOVB    (R0)+,R5        ; OTHERWISE GET TERMINATING CHARACTER
000352  020400                   CMP     R4,R0           ; REACHED THE END OF THE STRING ?
000354  101404                   BLOS    30$             ; IF LO, YES                    ;01
```

```
                                                                               D 14
PRSLIN      CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE               REFERENCES

LF.X2P  = 000000            #5-60
LN.CLO  = 000000            #5-60
LN.DUM  = 000005            #5-60
LN.LOA  = 000004            #5-60
LN.LOO  = 000003            #5-60
LN.OAU  = 000003            #5-60
LN.OFF  = 000001            #5-60
LN.ON   = 000000            #5-60
LN.OOP  = 000004            #5-60
LN.OPE  = 000001            #5-60
LN.RE[illegible]  = 000002            #5-60
LN.SER  = 000002            #5-60
LN.STA  = 000017            #5-60
LN.SUB  = 000360            #5-60
LN.TRI  = 000006            #5-60
LP$MPT  = 000010            #6-63
LP$MUX  = 000004            #6-63       7-121     7-133     10-315
LP$TRB  = 000002            #6-63
LP$UNT  = 000001            #6-63       7-143     7-151
LP$WCN  = 000040            #6-63       6-63      7-126     7-147     11-371
LP$WDV  = 000020            #6-63       6-63
LP$WLD  = 000360            #6-63
LP$WTR  = 000200            #6-63       6-63
LP$WUN  = 000100            #6-63       6-63      7-143     11-375
L$ADJ     000110            #5-63
L$BUF     000060            #5-63
L$CHN     000054            #5-63
L$COU     000114            #5-63
L$CTB     000053            #5-63
L$CTL     000002            #5-63      *7-132     11-373
L$DDM     000000            #5-63      *7-114     12-409
L$FLG     000064            #5-63
L$FLX     000070            #5-63
L$LEN     000124            #5-63
L$LTM     000070            #5-63
L$MSG     000056            #5-63
L$MTYP    000067            #5-63
L$NAM     000000            #5-63
L$NLEN  = 000020            #5-63       5-63      5-63
L$NOD     000112            #5-63
L$NXT     000044            #5-63
L$OPT     000062            #5-63
L$PAR     000120            #5-63
L$PDV     000055            #5-63       7-93
L$PFG     000041            #5-63       7-121    *7-126     7-133    *7-143     7-147    *7-151    *10-315    11-371
                              11-375
L$PLB     000116            #5-63
L$PRO     000067            #5-63
L$PVC     000004            #5-63
L$SCN     000020            #5-63
L$SCR     000072            #5-63
L$SLEN    000066            #5-63
```

```
                                                                               E 14
PRSLIN      CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 3
```

```
                                ;+
                                ;
                                ; ***  SPRAY - SPRAY A BYTE
                                ;
                                ; FUNCTION:
                                ;
                                ;   SPRAYS A SINGLE BYTE VALUE THROUGHOUT THE SPECIFIED AREA.
                                ;
                                ; INPUTS:
                                ;
                                ;   R0 - BYTE VALUE TO SPRAY
                                ;   R1 - ADDRESS TO BEGIN SPRAYING
                                ;   R2 - NUMBER OF BYTES TO SPRAY
                                ;
                                ; OUTPUTS:
                                ;
                                ;   R1 - ADDRESS OF NEXT BYTE AFTER SPRAYED AREA
                                ;
                                ; REGISTERS R3, R4, R5 ARE PRESERVED
                                ;-
                                ;
                                ; PROCEDURE BODY
                                ;
000270                          SPRAY::
000270  005302                  10$:    DEC     R2                      ;CHECK COUNT
000272  002402                          BLT     20$                     ;BR IF COUNT < 0
000274  110021                          MOVB    R0,(R1)+                ;MOVE A BYTE TO DESTINATION
000276  000774                          BR      10$

000300  000207                  20$:    RETURN


        000001                          .END
```

```
                                                .SBTTL  .MOMCL - PARSE MAXIMUM CLEARS
                                        ;+
                                        ; ** .MOMCL - PARSE MAXIMUM CLEARS
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $MXC = PARSED MAXIMUM CLEARS
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;-
000254                                  .MOMCL::
000254  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
000260  001112                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
000262  105767  000001G                         TSTB    .PNUMB+1        ; TEST HIGH BYTE
000266  001107                                  BNE     ILLNUM          ; BR IF  ILLEGAL NUMBER
000270  116767  000000G 000000G                 MOVB    .PNUMB,$MXC     ; MOVE VALUE TO PARSED AREA
000276  000207                                  RETURN

```

```
                                ;
                                ; REGISTERS:    USES R0-R2.
                                ;
                                ; CALLING SEQUENCE:  ENTER VIA CALLR FROM WITHIN A ACTION ROUTINE.
                                ;
                                ;-

000072                          $GTR50::
000072  010100                           MOV     R1,R0           ; COPY FIELD SIZE IN WORDS
000074  006300                           ASL     R0              ; X 2
000076  060100                           ADD     R1,R0           ; X 3 = SIZE IN CHARS
000100  026700  000000G                  CMP     .PSTCN,R0       ; CHECK SIZE
000104  101035                           BHI     REJECT          ; IF HI, REJECT TRANSITION
000106  010346                           MOV     R3,-(SP)        ; FREE UP A REGISTER
000110  016700  000000G                  MOV     .PSTPT,R0       ; POINT TO INPUT STRING
000114  010003                           MOV     R0,R3           ; COPY POINTER
000116  066703  000000G                  ADD     .PSTCN,R3       ; POINT TO CHARACTER AFTER LAST
000122  010346                           MOV     R3,-(SP)        ; SAVE ITS ADDRESS
000124  111346                           MOVB    (R3),-(SP)      ; SAVE IT
000126  112713  000077                   MOVB    #'?,(R3)        ; INSERT NON-RAD50 TERMINATOR
000132  066702  177646                   ADD     .TEMP,R2        ; POINT TO FIELD
000136  010203                           MOV     R2,R3           ; COPY FIELD POINTER
000140  016746  000000G                  MOV     .PSTCN,-(SP)    ; PUSH CHARACTER COUNT

000144  005022                  10$:     CLR     (R2)+           ; CLEAR A WORD
000146  005301                           DEC     R1              ; COUNTDOWN
000150  003375                           BGT     10$             ; UNTIL ENTIRE FIELD CLEARED

000152  010601                  20$:     MOV     SP,R1           ; ALLOW PERIODS
000154  004767  000000G                  CALL    .CAT5           ; GET RAD50 WORD (ALLOW '*')
000160  010123                           MOV     R1,(R3)+        ; STUFF IT INTO TEMPLATE,
000162  162716  000003                   SUB     #3,(SP)         ; TEST FOR END OF STRING
000166  003371                           BGT     20$             ; IF GT, MORE IN STRING

000170  005726                  30$:     TST     (SP)+           ; POP COUNT
000172  112636                           MOVB    (SP)+,@(SP)+    ; RESTORE TERMINATOR
000174  012603                           MOV     (SP)+,R3        ; RESTORE SCRATCH REGISTER
000176  000207                           RETURN

000200  062716  000002          REJECT:  ADD     #2,(SP)         ; REJECT TRANSITION
000204  000207                           RETURN


                                         .SBTTL  $CLERR - CLEAR PARSE ERROR
                                ;+
                                ; ** $CLERR - CLEAR PARSE ERROR
                                ;
                                ; INPUTS:        NONE.
                                ;
                                ; OUTPUT:        PARSE ERROR CLEARED.
                                ;
                                ;-

000206                          $CLERR::
000206  005067  177566                   CLR     .ERROR          ; CLEAR PARSE ERROR
000212  000207                           RETURN

```

```
                                                .SBTTL  $MOVIT - MOVE INPUT LINE INTO OUTPUT FILE WITHOUT EDITS
                                        ;+
                                        ; *** - $MOVIT - MOVE INPUT LINE INTO OUTPUT FILE WITHOUT EDITS
                                        ;
                                        ; INPUT:
                                        ;       $BUFI = INPUT LINE
                                        ;
                                        ; OUTPUT:
                                        ;       $OFDB+F.NRBD+2 = OUTPUT BUFFER ADDRESS
                                        ;
                                        ;-

000436                                  $MOVIT::
000436  012767  000000G 000000C                 MOV     #$BUFI,$OFDB+F.NRBD+2 ; POINT FDB AT OUTPUT BUFFER
000444  000207                                  RETURN                  ; RETURN TO CALLER
```

```
                                  .SBTTL  MACRO CALLS AND DATA AREA
                          ;
                          ; MACRO CALLS
                          ;
                                  .MCALL  CALLR

                          ;
                          ; GLOBAL DATA
                          ;
000000                    .ERROR:: .BLKW  1       ; PARSE ERROR MESSAGE ADDRESS
000002                    .FLAGS:: .BLKW  1       ; POINTER TO C.STS OF CURRENT TEMPLATE FOR FLAG SETTING
000004                    .TEMP::  .BLKW  1       ; ADDRESS OF CURRENT TEMPLATE
000006                    .COMMA:: .BLKW  1       ; COMMA COUNT FOR $PTOPT
000010                    .BUFF::  .BLKB  80.     ; ALTERNATE OM OUTPUT BUFFER
                          ;
                          ; LOCAL DATA
                          ;
000130                    BYTFLG: .BLKW   1       ; BYTE MODE FLAG FOR $PTNUM
```

```
                                                                        E  4
EVTTBL      CREATED BY  MACRO  ON 29-JUN-85 AT 05:16      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00
SYMBOL  VALUE                 REFERENCES

$KNEVT        000000 RG       #3-86
$$$B0       = 000001          #3-54       3-88       3-89       3-90       3-91       3-91       3-92       3-92       3-93
                               3-94       3-95       3-96       3-98       3-98       3-99       3-100      3-101      3-102
                               3-103      3-104      3-105      3-106      3-107      3-107

$$$B1       = 000002          #3-54       3-88       3-89       3-90       3-91       3-91       3-92       3-92       3-93
                               3-94       3-98       3-101      3-106      3-107      3-107

$$$B10      = 002000          #3-54       3-91       3-92       3-106      3-107
$$$B11      = 004000          #3-54       3-91       3-92       3-107
$$$B12      = 010000          #3-54       3-91       3-92       3-107
$$$B13      = 020000          #3-54       3-91       3-92       3-107
$$$B14      = 040000          #3-54       3-91       3-92       3-97       3-107
$$$B15      = 100000          #3-54       3-91       3-92       3-107
$$$B2       = 000004          #3-54       3-88       3-90       3-91       3-91       3-92       3-92       3-93       3-94
                               3-98       3-101      3-106      3-107      3-107

$$$B3       = 000010          #3-54       3-88       3-91       3-91       3-92       3-93       3-98       3-101      3-106
                               3-107      3-107

$$$B4       = 000020          #3-54       3-88       3-91       3-92       3-93       3-106      3-107      3-107
$$$B5       = 000040          #3-54       3-88       3-91       3-92       3-93       3-106      3-107
$$$B6       = 000100          #3-54       3-88       3-91       3-92       3-106      3-107
$$$B7       = 000200          #3-54       3-88       3-91       3-92       3-106      3-107
$$$B8       = 000400          #3-54       3-88       3-91       3-92       3-102      3-106      3-107
$$$B9       = 001000          #3-54       3-88       3-91       3-92       3-102      3-106      3-107
$$$W0       = 177777          #3-88       3-88      #3-88       3-88      #3-88       3-88      #3-88       3-88      #3-88
                               3-88      #3-88       3-88      #3-88       3-88      #3-88       3-88      #3-88       3-88
                              #3-88       3-88      #3-88       3-88      #3-89       3-89      #3-89       3-89      #3-89
                               3-89      #3-90       3-90      #3-90       3-90      #3-90       3-90      #3-90       3-90
                              #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-91
                               3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91
                              #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-91
                               3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-92       3-92
                              #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-92
                               3-92      #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-92       3-92
                              #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-92
                               3-92      #3-92       3-92      #3-92       3-92      #3-93       3-93      #3-93       3-93
                              #3-93       3-93      #3-93       3-93      #3-93       3-93      #3-93       3-93      #3-93
                               3-93      #3-94       3-94      #3-94       3-94      #3-94       3-94      #3-94       3-94
                              #3-95       3-95      #3-95       3-95      #3-96       3-96      #3-96       3-96      #3-97
                               3-97      #3-97       3-97      #3-98       3-98      #3-98       3-98      #3-99       3-99
                              #3-99       3-99      #3-100      3-100     #3-100      3-100     #3-101      3-101     #3-101
                               3-101     #3-101      3-101     #3-101      3-101     #3-101      3-101     #3-102      3-102
                              #3-102      3-102     #3-102      3-102     #3-102      3-102     #3-103      3-103     #3-103
                               3-103     #3-104      3-104     #3-104      3-104     #3-105      3-105     #3-105      3-105
                              #3-106      3-106     #3-106      3-106     #3-106      3-106     #3-106      3-106     #3-106
                               3-106     #3-106      3-106     #3-106      3-106     #3-106      3-106     #3-106      3-106
                              #3-106      3-106     #3-106      3-106     #3-106      3-106     #3-107      3-107     #3-107
                               3-107     #3-107      3-107     #3-107      3-107     #3-107      3-107     #3-107      3-107
                              #3-107      3-107     #3-107      3-107     #3-107      3-107     #3-107      3-107     #3-107
                               3-107     #3-107      3-107     #3-107      3-107     #3-107      3-107     #3-107      3-107
                              #3-107      3-107     #3-107      3-107
$$$W1       = 000037          #3-88       3-88      #3-89       3-89      #3-90       3-90      #3-91       3-91      #3-91
                               3-91      #3-91       3-91      #3-91       3-91      #3-91       3-91      #3-92       3-92
                              #3-92       3-92      #3-92       3-92      #3-92       3-92      #3-93       3-93      #3-94
```

```
                                                                        F  4
EVTTBL      CREATED BY  MACRO  ON 29-JUN-85 AT 05:16      PAGE 2
```

```
   130 000000                        PRMPT$  <"Clear object options: ">,           P$CLOB, <:>
   131 000000                        PRMPT$  <"Clear timer (sec): ">,              P$CLT,  <:>
   132 000000                        PRMPT$  <"Clear trace options: ">,            P$CLTR, <:>
   133 000000                        PRMPT$  <"Clock mode: ">,                     P$CLK,  <:>
   134 000000                        PRMPT$  <"Command: ">,                        P$COM,  <:>
   135 000000                        PRMPT$  <"Component type: ">,                 P$PEN,  <:>
   136 000000                        PRMPT$  <"Controller mode: ">,                P$CON,  <:>
   137 000000                        PRMPT$  <"Copies: ">,                         P$COP,  <:>
   138 000000                        PRMPT$  <"Cost: ">,                           P$COS,  <:>
   139 000000                        PRMPT$  <"Counter timer (sec): ">,            P$CTM,  <:>
   140 000000                        PRMPT$  <"CPU type: ">,                       P$CPU,  <:>
   141 000000                        PRMPT$  <"CSR address: ">,                    P$CSR,  <:>
   142 000000                        PRMPT$  <"Dead polling ratio: ">,             P$DEA,  <:>
   143 000000                        PRMPT$  <"Dead threshold: ">,                 P$DDTH, <:>
   144 000000                        PRMPT$  <"Dead timer (msec): ">,              P$DDT,  <:>
   145 000000                        PRMPT$  <"Default data (bytes): ">,           P$BDF,  <:>
   146 000000                        PRMPT$  <"Default window (bytes): ">,         P$WDF,  <:>
   147 000000                        PRMPT$  <"Delay factor: ">,                   P$DFA,  <:>
   148 000000                        PRMPT$  <"Delay timer (msec): ">,             P$DLT,  <:>
   149 000000                        PRMPT$  <"Delay weight: ">,                   P$DWE,  <:>
   150 000000                        PRMPT$  <"Destination id:">,                  P$DST,  <:>
   151 000000                        PRMPT$  <"Destination node: ">,               P$DES,  <:>
   152 000000                        PRMPT$  <"Device spec: ">,                    P$DEV,  <:>
   153 000000                        PRMPT$  <"Diagnostic file: ">,                P$DIA,  <:>
   154 000000                        PRMPT$  <"DTE address: ">,                    P$DTE,  <:>
   155 000000                        PRMPT$  <"DTE number: ">,                     P$NUM,  <:>
   156 000000                        PRMPT$  <"Dump address: ">,                   P$DUA,  <:>
   157 000000                        PRMPT$  <"Dump count: ">,                     P$DUC,  <:>
   158 000000                        PRMPT$  <"Dump file: ">,                      P$DUM,  <:>
   159 000000                        PRMPT$  <"Dump option: ">,                    P$DUMO, <:>
   160 000000                        PRMPT$  <"Duplex: ">,                         P$DUP,  <:>
   161 000000                        PRMPT$  <"Dying base: ">,                     P$BSD,  <:>
   162 000000                        PRMPT$  <"Dying increment: ">,                P$IND,  <:>
   163 000000                        PRMPT$  <"Dying threshold: ">,                P$DYTH, <:>
   164 000000                        PRMPT$  <"Event identification: ">,           P$EVE,  <:>
   165 000000                        PRMPT$  <"File name: ">,                      P$FNAM, <:>
   166 000000                        PRMPT$  <"Group name: ">,                     P$GRP,  <:>
   167 000000                        PRMPT$  <"Group number: ">,                   P$GNM,  <:>
   168 000000                        PRMPT$  <"Hello timer (sec): ">,              P$HTM,  <:>
   169 000000                        PRMPT$  <"Hardware address: ">,               P$HAD,  <:>
   170 000000                        PRMPT$  <"Hardware option: ">,                P$HAO,  <:>
   171 000000                        PRMPT$  <"Hexadecimal password: ">,           P$HPAS, <:>
   172 000000                        PRMPT$  <"Holdback timer (msec): ">,          P$HBT,  <:>
   173 000000                        PRMPT$  <"Host node identification: ">,       P$HOS,  <:>
   174 000000                        PRMPT$  <"Identification: ">,                 P$IDE,  <:>
   175 000000                        PRMPT$  <"Inactive base: ">,                  P$BSI,  <:>
   176 000000                        PRMPT$  <"Inactive increment: ">,             P$INI,  <:>
   177 000000                        PRMPT$  <"Inactive threshold: ">,             P$INTH, <:>
   178 000000                        PRMPT$  <"Inactivity timer (sec): ">,         P$IAT,  <:>
   179 000000                        PRMPT$  <"Incoming timer (sec): ">,           P$ITI,  <:>
   180 000000                        PRMPT$  <"Length of test data: ">,            P$LEN,  <:>
   181 000000                        PRMPT$  <"Line identification: ">,            P$LNID, <:>
   182 000000                        PRMPT$  <"Line owner: ">,                     P$LOWN, <:>
   183 000000                        PRMPT$  <"Line protocol: ">,                  P$PRO,  <:>
   184 000000                        PRMPT$  <"Line state: ">,                     P$LIST, <:>
   185 000000                        PRMPT$  <"Load image input file: ">,          P$LOA,  <:>
   186 000000                        PRMPT$  <"Load options: ">,                   P$LDOP, <:>
```

```
    187 000000                  PRMPT$  <'Location in memory: '>,               P$LOC,  <:>
    188 000000                  PRMPT$  <'Logging sink type: '>,                P$LGID, <:>
    189 000000                  PRMPT$  <'Logging state: '>,                    P$LOST, <:>
    190 000000                  PRMPT$  <'Loop component: '>,                   P$LOO,  <:>
    191 000000                  PRMPT$  <'Loop help type: '>,                   P$LHE,  <:>
    192 000000                  PRMPT$  <'Maximum address: '>,                  P$MAD,  <:>
    193 000000                  PRMPT$  <'Maximum Area Number: '>,              P$AREA, <:>
    194 000000                  PRMPT$  <'Maximum block (bytes): '>,            P$LMB,  <:>
    195 000000                  PRMPT$  <'Maximum Broadcast Routers: '>,        P$MBR,  <:>
    196 000000                  PRMPT$  <'Maximum Broadcast NonRouters: '>,     P$MBN,  <:>
    197 000000                  PRMPT$  <'Maximum buffers: '>,                  P$MBU,  <:>
    198 000000                  PRMPT$  <'Maximum circuits: '>,                 P$MLN,  <:>
    199 000000                  PRMPT$  <'Maximum clears: '>,                   P$LMX,  <:>
    200 000000                  PRMPT$  <'Maximum controllers: '>,              P$MXCO, <:>
    201 000000                  PRMPT$  <'Maximum cost: '>,                     P$MCO,  <:>
    202 000000                  PRMPT$  <'Maximum data (bytes): '>,             P$CMB,  <:>
    203 000000                  PRMPT$  <'Maximum hops: '>,                     P$MHO,  <:>
    204 000000                  PRMPT$  <'Maximum lines: '>,                    P$MXLI, <:>
    205 000000                  PRMPT$  <'Maximum links: '>,                    P$MLK,  <:>
    206 000000                  PRMPT$  <'Maximum recall count: '>,             P$MXR,  <:>
    207 000000                  PRMPT$  <'Maximum receive buffers: '>,          P$MRB,  <:>
    208 000000                  PRMPT$  <'Maximum resets: '>,                   P$RMX,  <:>
    209 000000                  PRMPT$  <'Maximum restarts: '>,                 P$SMX,  <:>
    210 000000                  PRMPT$  <'Maximum retransmits: '>,              P$MRT,  <:>
    211 000000                  PRMPT$  <'Maximum Routers: '>,                  P$MXRT  <:>
    212 000000                  PRMPT$  <'Maximum transmit count: '>,           P$MXB,  <:>
    213 000000                  PRMPT$  <'Maximum visits: '>,                   P$MVI,  <:>
    214 000000                  PRMPT$  <'Maximum window (blks): '>,            P$MXW,  <:>
    215 000000                  PRMPT$  <'Module identification: '>,            P$MOID, <:>
    216 000000                  PRMPT$  <'Multipoint ratio type: '>,            P$MUL,  <:>
    217 000000                  PRMPT$  <'Network name: '>,                     P$NET,  <:>
    218 000000                  PRMPT$  <'Node identification: '>,              P$NID,  <:>
    219 000000                  PRMPT$  <'Node name: '>,                        P$NAM,  <:>
    220 000000                  PRMPT$  <'Executor state: '>,                   P$NOST, <:>
    221 000000                  PRMPT$  <'Executor Node Type: '>,               P$ETY,  <:>
    222 000000                  PRMPT$  <'Number of test messages: '>,          P$COU,  <:>
    223 000000                  PRMPT$  <'Object identification: '>,            P$OBJ,  <:>
    224 000000                  PRMPT$  <'Object name: '>,                      P$OBNA, <:>
    225 000000                  PRMPT$  <'Object number: '>,                    P$OID,  <:>
    226 000000                  PRMPT$  <'Object user: '>,                      P$OBUS, <:>
    227 000000                  PRMPT$  <'Object verification: '>,              P$OBVE, <:>
    228 000000                  PRMPT$  <'Originating Queue Limit: '>,          P$OQL,  <:>
    229 000000                  PRMPT$  <'Outgoing timer (sec): '>,             P$OTI,  <:>
    230 000000                  PRMPT$  <'Output file/device: '>,               P$TO,   <:>
    231 000000                  PRMPT$  <'Partition name: '>,                   P$PAR,  <:>
    232 000000                  PRMPT$  <'Password: '>,                         P$PAS,  <:>
    233 000000                  PRMPT$  <'Physical address: '>                  P$PHA,  <:>
    234 000000                  PRMPT$  <'Physical option: '>                   P$PHO,  <:>
    235 000000                  PRMPT$  <'Polling state: '>,                    P$PST,  <:>
    236 000000                  PRMPT$  <'Priority: '>,                         P$PRI,  <:>
    237 000000                  PRMPT$  <'Process name: '>,                     P$PID,  <:>
    238 000000                  PRMPT$  <'Router Priority: '>,                  P$RTPR  <:>
    239 000000                  PRMPT$  <'Recall timer (sec): '>,               P$RET,  <:>
    240 000000                  PRMPT$  <'Receive buffers: '>,                  P$BFQ,  <:>
    241 000000                  PRMPT$  <'Reset timer (sec): '>,                P$RST,  <:>
    242 000000                  PRMPT$  <'Restart timer (sec): '>,              P$RRT,  <:>
    243 000000                  PRMPT$  <'Retransmit factor: '>,                P$RFA,  <:>
```

```
                                        .SBTTL  .NONAM - NODE NAME ACTION ROUTINE
                                ;+
                                ; *** - .NONAM - NODE NAME ACTION ROUTINE
                                ;
                                ; INPUT:
                                        .PSTPT = ADDRESS OF NODE NAME
                                        .PSTCN = COUNT OF CHARS IN NODE NAME
                                ;
                                ; OUTPUT:
                                        $NODNA = RAD50 NODE NAME (2 WORDS)
                                        R0,R1 = DESTROYED
                                        THE TRANSITION IS ACCEPTED OR REJECTED
                                ;
                                ;-

000224                          .NONAM::
000224  004767  000144                  CALL    NODNAM                  ; CONVERT NODE NAME
000230  103430                          BCS     10$                     ; IF CS, ERROR
000232  026767  000000G 000000G         CMP     $NODAD,$NTADD           ; IS THIS THE EXECUTOR NODE?
000240  001014                          BNE     5$                      ; BR IF NO
000242  026767  000000G 000000G         CMP     $NODNA,$NTNAM           ; IS THIS THE EXECUTOR NODE NAME?
000250  001004                          BNE     3$                      ; BR IF NO ERROR
000252  026767  000002G 000002G         CMP     $NODNA+2,$NTNAM+2 ; REALLY?
000260  001430                          BEQ     30$                     ; BR IF YES .. OKAY
000262  012767  000105' 000000G 3$:     MOV     #ERREXE,$ERROR          ; ELSE, ILLEGAL TO CHANGE EXECUTOR NAME
000270  000422                          BR      25$                     ; BR TO INDICATE ERROR
000272  026767  000000G 000000G 5$:     CMP     $NODNA,$NTNAM           ; IS THIS THE EXECUTOR NODE NAME ?
000300  001004                          BNE     10$                     ; IF NE, NO .. OKAY
000302  026767  000002G 000002G         CMP     $NODNA+2,$NTNAM+2 ; REALLY ?
000310  001407                          BEQ     20$                     ; IF EQ, YES .. ERROR
000312  004767  000736          10$:    CALL    FNDNAM                  ; TRY TO FIND NODE NAME IN REMOTE BLOCKS
000316  103411                          BCS     30$                     ; IF CS, CAN'T .. OKAY
000320  026067  000010  000000G         CMP     C.ADD(R0),$NODAD ; ELSE, IS THIS THE SAME NODE ?
000326  001405                          BEQ     30$                     ; IF EQ, YES .. OKAY
000330  012767  000000' 000000G 20$:    MOV     #ERRNAM,$ERROR          ; ELSE, NAME ALREADY IN USE
000336  062716  000002          25$:    ADD     #2,(SP)                 ; REJECT TRANSITION
000342  000207                  30$:    RETURN
```

```
                                                                     E 8
PRSACT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:17       PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE             REFERENCES

CM.DEF  = ******  GX       11-330
CM.LIS  = ******  GX       11-328      14-410
CO.EXE  = ******  GX       11-332      14-412
CS.REM  = ******  GX       12-358      16-473
C.ADD     000010          #4-76        8-169       12-360      14-419      15-450
C.NAM     000004          #4-75       16-475       16-477
C.STS   = ******  GX       12-358      16-473
ERRADD    000041 R        #4-64       12-362
ERREXE    000105 R        #4-65        8-161       11-334      14-421
ERRILC  = ******  GX       14-401
ERRILN  = ******  GX       11-285      11-336      12-365
ERRNAM    000000 R        #4-63        8-171        9-195
ERRNOD  = ******  GX       14-423
EX$ERR  = 000002          #4-62        4-63         4-64        4-65
FNDNAM    001254 R         8-167       9-193       14-414      15-448     #16-468
ILLNUM    001024 R        #12-364     13-383       17-496      17-501      17-504      18-522      18-524      18-526
NODNAM    000374 R         8-153       9-191      #10-213      14-403      15-439
START   = ******  GX      *5-95
$BL.AU    000206 RG       #6-117
$BL.A3    000200 RG       #6-115
$BL.MA    000156 RG       #5-93
$BL.MN    000150 RG       #5-91
$BSZ    = ******  GX      *17-498
$CAT5   = ******  GX       10-217      10-221
$CMAND  = ******  GX       11-328      11-330       14-410
$CMDPT  = ******  GX      *7-134
$COMP   = ******  GX      *11-332     *14-412
$ERROR  = ******  GX      *8-161      *8-171       *9-195     *11-285     *11-334     *11-336     *12-362     *12-365      14-401
                          *14-421     *14-423
$NODAD  = ******  GX       8-155       8-169      *11-241      11-264     *11-276     *11-282     *11-312      11-319     *11-325
                           11-326     12-360      *14-419     *15-445     *15-450
$NODNA  = ******  GX       8-157       8-159        8-163       8-165     *10-214     *10-218     *10-222      14-406      14-408
                           15-441     15-443       16-475      16-477
$NTADD  = ******  GX       8-155      11-257       11-282      11-309      11-321      11-325      11-326      15-445
$NTNAM  = ******  GX       8-157       8-159        8-163       8-165      14-406      14-408      15-441      15-443
$RTM    = ******  GX      *13-382
$TMLST  = ******  GX       12-355      16-469
$WSZ    = ******  GX      *18-523
.BL.AU    000174 RG       #6-113
.EXADD    000764 RG       #12-354
.EXNAM    000344 RG       #9-190
.EXRTM    001040 RG       #13-379
.HLPAC    000216 RG       #7-133
.NONAM    000224 RG       #8-152
.PFLAG  = ******  GX      *5-92       *5-94        *6-116      *6-118
.PNUMB  = ******  GX       11-261      11-307      13-382      17-497      18-523      18-525
.PNUMH  = ******  GX       11-259      11-305      17-495      18-521
.PRMDA    001322 RG       #17-494
.PRMWI    001370 RG       #18-520
.PSTCN  = ******  GX       10-223
.PSTPT  = ******  GX       5-95        10-215
.RNADD    000612 RG       #11-304
```

```
                                                                     F 8
PRSACT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:17       PAGE 2
```

```
                                                .SBTTL  NEWPVC - Check circuit syntax for new X.25 PVC
                                        ;+
                                        ; *** NEWPVC - Check circuit syntax for new X.25 PVC
                                        ;
                                        ; FUNCTION
                                        ;       If the circuit name has 'X25-' as a prefix, then
                                        ;       indicate in the context area that a new circuit has been requested.
                                        ;
                                        ; INPUTS
                                        ;       R1 - Address of circuit name as counted string
                                        ;       R5 - Address of context area
                                        ;
                                        ; OUTPUTS
                                        ;       Carry set if not a possible new circuit
                                        ;       Otherwise:
                                        ;               L$TPT(R5) set to -1 (indicates X.25 circuit, and
                                        ;               prevents NXTCIR trying to find it in the PVC list)
                                        ;               L$SLT(R5) set to 0 (indicates not yet extant)
                                        ;               L$NAM(R5) set to circuit id
                                        ;       R0 destroyed.
                                        ;-

000460                                  NEWPVC:
000460                                          SAVRG   <R1,R2>                 ; Save work registers
000464  017700  000000G                         MOV     @PSIPT,R0               ; Check we have PSI             ; RJK05
000470  001426                                  BEQ     20$                     ; Branch if not
000472  032760  000010  000000                  BIT     #HF$GWY,H$FLG(R0)       ; Check we are a gateway
000500  001422                                  BEQ     20$                     ; Branch if not

000502  112100                                  MOVB    (R1)+,R0                ; Get length of circuit-id
000504  020027  000004                          CMP     R0,#XPREFL              ; Long enough to include prefix?
000510  101416                                  BLOS    20$                     ; Branch if not
000512  012700  000004                          MOV     #XPREFL,R0
000516  012702  000000'                         MOV     #XPREF,R2               ; Point to PVC prefix string
000522  122122                          10$:    CMPB    (R1)+,(R2)+             ; See if prefix present
000524  001010                                  BNE     20$                     ; Branch if not
000526                                          SOB     R0,10$

000532  012765  177777  000050                  MOV     #-1,L$TPT(R5)           ; Flag circuit as X.25 circuit
000540  005065  000042                          CLR     L$SLT(R5)               ; Avoid confusion with existing circs.

000544  005727                                  TST     (PC)+                   ; Clear carry (success) and skip
000546  000261                          20$:    SEC
000550                                          RESRG   <R2,R1>                 ; Restore work registers
000554                                          RETURN

                                                .ENDC ; NDF R$$11S
```

```
PRSCIR - PARSE A CIRCUIT-ID     MACRO V05.03b Saturday 29-Jun-85 13:20  Page 16-4
Symbol table

N$$NCT= 000001          SF$ANM  000001          S$$YSZ= 007600          ZF.DVP= 100000          Z.DAT   000016
N$$PEM= 000001          SF$DNM  000002          S.COST  000001          ZF.INI= 040000          Z.DSP   000000
PDVAD   001334R         SF.ACT= 000200          S.FLG   000000          ZF.KMX= 000020          Z.FLG   000010
PDVID = ****** GX       SF.ENA= 000100          S.LEN   000004          ZF.LLC= 000004          Z.LEN = 000016
PDVTA = ****** GX       SF.LPB= 000004          S.NMST  000002          ZF.LMC= 000100          Z.LLN   000006
PRSCIR  000004RG        SF.MFL= 000040          S.OWNR  000003          ZF.MAN= 020000          Z.MAP   000020
PSIPT = ****** GX       SF.PAC= 000020          T$$KMG= 000000          ZF.MFL= 000010          Z.NAM   000004
P$$P45= 000000          SF.REA= 000010          T$$MIN= 000000          ZF.MTM= 000400          Z.PCB   000012
P$$WRD= 000000          SF.SER= 000001          V$$CTR= 001000          ZF.MUX= 000040          Z.SCH   000007
Q$$OPT= 000010          SF.SVC= 000002          XPREF   000000R         ZF.PSE= 002000          $FNPVC= ****** GX
R$R0  = ****** GX       SF.UNL= 000040          XPREFL= 000004          ZF.SLI= 010000          $HEADR= ****** GX
R$R1  = ****** GX       SLTMA = ****** GX       X$$DBT= 000000          ZF.TIM= 000200          $MUL  = ****** GX
R$$DER= 000000          SLTNM = ****** GX       ZF.COU= 001000          ZF.X3P= 000000          $SAVAL= ****** GX
R$$K11= 000001          SPACE = 000040          ZF.DDM= 000001          ZS.ASN= 100000          $$$R  = 000010
R$$SND= 000000          STOCIR  000402RG        ZF.DIA= 004000          ZS.BSY= 140000          $$$SYS= 004374
R$$11M= 000000          S$$WRG= 000000          ZF.DLC= 000002          Z.AVL   000014          $$$SYX= 000000


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001434    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 224
Work  file writes: 158
Size of work file: 31874 Words  ( 125 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system. RSX-11M/PLUS

Elapsed time: 00:00:38.03
SY:PRSCIR.V2,[135,134]PRSCIR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[135,10]PRSCIR
```

```
                                                          F 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 1
```

```
    168                                                 .SBTTL  .EXCOS - SET MAXIMUM LINE COST IN NETWORK
    169                                         ;+
    170                                         ; *** - .EXCOS - SET MAXIMUM LINE COST IN NETWORK
    171                                         ;
    172                                         ; INPUTS:
    173                                         ;       .PNUMB = THE MAXIMUM LINE COST IN NETWORK
    174                                         ;
    175                                         ; OUTPUTS:
    176                                         ;       $MAXC = THE MAXIMUM LINE COST IN NETWORK
    177                                         ;       THE  TRANSITION IS ACCEPTED/REJECTED.
    178                                         ;
    179                                         ;-
    180
    181 000240                                  .EXCOS::
    182 000240  016767  000000G 000000G                 MOV     .PNUMB,$MAXC    ; STORE THE MAX. LINE COST
    183 000246  001404                                  BEQ     10$             ; IF EQ, ILLEGAL VALUE
    184 000250  026727  000000G 000000G                 CMP     .PNUMB,#MAXCST  ; IS THE VALUE IN RANGE ?
    185 000256  101405                                  BLOS    20$             ; IF LOS, YES .. OKAY
    186 000260  062716  000002                  10$:    ADD     #2,(SP)         ; ELSE, REJECT TRANSITION
    187 000264  012767  000000G 000000G                 MOV     #ERRILN,$ERROR  ; INVALID NUMBER
    188 000272  000207                          20$:    RETURN
```

```
Symbol table

CS.ROU= ****** GX       $EAHOP= ****** GX       $LINKS= ****** GX       .EAMAX  000624RG        .EXNC   000360RG
C.ADD = 000004          $EAMAX= ****** GX       $MAXC = ****** GX       .EXBEA  001002RG        .EXNN   000274RG
C.STS = ****** GX       $EBEA = ****** GX       $MAXH = ****** GX       .EXBRT  000750RG        .EXOUT  001176RG
ERRILN= ****** GX       $EBRT = ****** GX       $MVASC= ****** GX       .EXCOS  000240RG        .EXRET  001236RG
ERRPNA= ****** GX       $EINA = ****** GX       $NCOU = ****** GX       .EXHOP  000204RG        .EXSB2  000552RG
MAXCST= ****** GX       $EINC = ****** GX       $NN   = ****** GX       .EXHOS  000476RG        .EXSEG  001044RG
MAXHOP= ****** GX       $EOUT = ****** GX       $NODAD= ****** GX       .EXHO1  000466RG        .EXSUB  000544RG
MAXNN = ****** GX       $ERET = ****** GX       $NTADD= ****** GX       .EXID   000000RG        .PFLAG= ****** GX
NF.LV2= ****** GX       $ERROR= ****** GX       $OPRND= ****** GX       .EXID1  000064RG        .PNUMB= ****** GX
OE.NN = ****** GX       $ESEG = ****** GX       $TMLST= ****** GX       .EXINA  001076RG        .PNUMH= ****** GX
STRMX = 000040          $ESUB = ****** GX       .EACOS  000644RG        .EXINT  001136RG        .PSTCN= ****** GX
$CNFIG= ****** GX       $HOST = ****** GX       .EAHOP  000666RG        .EXLNK  000152RG        .PSTPT= ****** GX
$EACOS= ****** GX       $ID   = ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001276    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 122 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:12.00
SY:PRSEXE.V2,[132,134]PRSEXE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSEXE
```

F 12

```
PRSEXE     CREATED BY  MACRO  ON 29-JUN-85 AT 05:18     PAGE 1
```

```

000356  120305            20$:    CMPB    R3,R5           ; IS THE CHARACTER A LEGAL TERMINATOR ?
000360  001403                    BEQ     40$             ; IF EQ, YES - RETURN SUCCESS CODE
000362  000261                    SEC                     ; OTHERWISE INDICATE AN ERROR
000364  000402                    BR      50$             ;

000366  010616            30$:    MOV     SP,(SP)         ; MARK THE END OF THE STRING

000370  000241            40$:    CLC                     ; INDICATE SUCCESSFUL CONVERSION
000372  012602            50$:    MOV     (SP)+,R2        ; RESTORE REGISTERS
000374  012605                    MOV     (SP)+,R5        ; SET EXIT CONDITION CODES
000376                            RETURN                  ; RETURN TO CALLER
```

E 14

```
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23     PAGE 3
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE              REFERENCES

L$SLT     000042          #5-63
L$SNM     000046          #5-63
L$SYL     000122          #5-63
L$TNM     000052          #5-63
L$TPT     000050          #5-63
L$TRB     000040          #5-63
L$TSZ     000070          #5-63
L$TYP     000063          #5-63
L$UNT     000003          #5-63       *7-152       11-377
L.COST    000015          #5-60
L.CTL     000012          #5-60        11-373
L.CVA     177776          #5-60
L.DDM     000002          #5-60        11-369
L.DDS     000004          #5-60
L.DLC     000003          #5-60
L.DLM     000006          #5-60
L.DLS     000010          #5-60
L.FLG     000000          #5-60
L.KRBA    000016          #5-60
L.LEN   = 000022          #5-60
L.MPF     000022          #5-60
L.NMST    000020          #5-60
L.NSTA    000014          #5-60
L.OWNR    000021          #5-60
L.UNT     000013          #5-60        11-377
ME.IID    177767            7-163      11-383
ME.UCO    177770            11-386
N$$VCT  = ******            11-387
PDVAD     000662 R          10-311     11-361       #12-408
PDVID   = ******  GX        12-410
PDVTA   = ******  GX        12-413
PRSLIN    000000 RG       #7-90
R$R0    = ******  GX      *7-160      *7-163
R$R1    = ******  GX      *7-99
R$$RTR  = ******            7-101      11-341
R$$11D  = ******            5-59
R$$11M  = 000000            5-59
R$$11S  = ******            5-59
SF.ACT  = 000200          #5-60
SF.ENA  = 000100          #5-60
SF.LPB  = 000004          #5-60
SF.MFL  = 000040          #5-60
SF.PAC  = 000020          #5-60
SF.REA  = 000010          #5-60
SF.SER  = 000001          #5-60
SF.SVC  = 000002          #5-60
SF.UNL  = 000040          #5-60
SLTMA   = ******  GX        11-367
SLTNM   = ******  GX        11-366
S$$BAS  = ******            7-101      11-341
S.COST    000001          #5-60
S.FLG     000000          #5-60
```

F 14

```
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23     PAGE 4
```

F 15

```
PRSLOG - PARSE LOGGING ACTION R MACRO V05.03b Saturday 29-Jun-85 05:19  Page 10-1
Symbol table

CKEVE   000000R           CKEVEN  000204RG        SPRAY   000270RG        $EVBEG= ****** GX       .PNUMB= ****** GX
CKEVEB  000064RG          CKEVER  000224RG        $CLASS= ****** GX       $EVENT= ****** GX       .PNUMH= ****** GX
CKEVEC  000114RG

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000302    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 147 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.04
SY:PRSLOG.V2,[132,134]PRSLOG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSLOG
```

F 15

```
PRSLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 1
```

```
                                              .SBTTL  .MOMRE - PARSE MAXIMUM RESETS
                                      ;+
                                      ; ** .MOMRE - PARSE MAXIMUM RESETS
                                      ;
                                      ; INPUTS        .TPARS VARIABLES.
                                      ;
                                      ; OUTPUT        $MRE = PARSED MAXIMUM RESETS
                                      ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                      ;-
000300                                .MOMRE::
000300  005767  000000G                       TST     .PNUMH          ; HIGH ORDER WORD
000304  001100                                BNE     ILLNUM          ; REJECT IF TOO HIGH
000306  105767  000001G                       TSTB    .PNUMB+1        ; TEST HIGH BYTE
000312  001075                                BNE     ILLNUM          ; BR IF  ILLEGAL NUMBER
000314  116767  000000G 000000G               MOVB    .PNUMB,$MRE     ; MOVE VALUE TO PARSED AREA
000322  000207                                RETURN

```

```
    183         000001                          .END
```

```
                                        ;+
                                        ; *** - $MOVST - MOVE FIRST FIELD OF INPUT BUFFER INTO OUTPUT BUFFER
                                        ;
                                        ; INPUT:
                                        ;       R4 = REST OF LINE AFTER FIRST FIELD
                                        ;
                                        ; OUTPUT:
                                        ;       $OBUF = CURRENT POSITION IN OUTPUT BUFFER
                                        ;
                                        ;-

000446                                  $MOVST::
000446  012700  000000G                         MOV     #$BUFR,R0       ; POINT R0 AT OUTPUT BUFFER
000452  012701  000000G                         MOV     #$BUFI,R1       ; POINT R1 AT INPUT BUFFER
000456  010402                                  MOV     R4,R2           ; COPY ADDRESS OF REST OF INPUT LINE
000460  160102                                  SUB     R1,R2           ; CALCULATE LENGTH OF FIRST + 1
000462  005202                                  INC     R2              ;                           ...
000464  004767  000000G                         CALL    $MVASC          ; MOVE INPUT STRING TO OUTPUT BUFFER
000470  010067  177310                          MOV     R0,$OBUF        ; SAVE CURRENT POSITION IN OUTPUT BUFFER
000474  000207                                  RETURN                  ; RETURN TO CALLER
```

```
                                         .SBTTL  $PTNUM - PUT NUMBER INTO TEMPLATE (WORD)
                                         .SBTTL  $PTNMB - PUT NUMBER INTO TEMPLATE (BYTE)
                                 ;+
                                 ; ** $PTNUM - PUT NUMBER INTO TEMPLATE (WORD)
                                 ; ** $PTNMB - PUT NUMBER INTO TEMPLATE (BYTE)
                                 ;
                                 ; INPUTS:       .TEMP = TEMPLATE ADDRESS
                                 ;               R2 = OFFSET IN TEMPLATE
                                 ;               R1 =   0 FOR DECIMAL OUTPUT
                                 ;                    <>0 FOR OCTAL OUTPUT
                                 ;
                                 ; OUTPUT:       VALUE LOADED INTO OUTPUT BUFFER.
                                 ;
                                 ; REGISTERS:    R0-R2 ARE CORRUPTED.
                                 ;
                                 ;-
                                         .ENABL  LSB

000132                           $PTNMB::
000132  010667  177772                   MOV     SP,BYTFLG          ; SET UP BYTE MODE
000136  000402                           BR      10$                ; JOIN COMMON CODE

000140                           $PTNUM::
000140  005067  177764                   CLR     BYTFLG             ; SET UP WORD MODE
000144  004767  000144           10$:    CALL    CHKOPT             ; OUTPUT COMMAS FOR ANY MISSING PARAMS
000150  012746  000000G                  MOV     #$CNVD,-(SP)       ; ASSUME DECIMAL CONVERSION
000154  005701                           TST     R1                 ; CHECK RADIX
000156  001402                           BEQ     20$                ; IF EQ, DECIMAL
000160  012716  000000G                  MOV     #$CNVO,(SP)        ; ELSE POINT TO OCTAL ROUTINE
000164  066702  177614           20$:    ADD     .TEMP,R2           ; POINT TO SOURCE FIELD
000170  005767  177734                   TST     BYTFLG             ; CHECK FOR BYTE FIELD
000174  001404                           BEQ     30$                ; IF EQ, WORD
000176  111201                           MOVB    (R2),R1            ; GET BYTE
000200  042701  177400                   BIC     #^C377,R1          ; NO SIGN EXTEND
000204  000401                           BR      40$                ; JOIN COMMON CODE
000206  011201                   30$:    MOV     (R2),R1            ; GET WORD
000210  016700  000000G          40$:    MOV     $OBUF,R0           ; POINT TO OUTPUT BUFFER
000214  004736                           CALL    @(SP)+             ; DO THE CONVERSION
000216  010067  000000G                  MOV     R0,$OBUF           ; UPDATE BUFFER POINTER
000222  000207                           RETURN

                                         .DSABL  LSB

                                         .SBTTL  $PTR50 - PUT RAD50
                                 ;+
                                 ; ** $PTR50 - PUT RAD50
                                 ;
                                 ; INPUTS:       .TEMP = TEMPLATE ADDRESS
                                 ;               R2 = OFFSET IN TEMPLATE
                                 ;               R1 = SIZE OF FIELD IN WORDS.
                                 ;
                                 ; OUTPUT:       THE RAD50 FIELD IS OUTPUT.
                                 ;
                                 ; REGISTERS:    USES R0-R2.
                                 ;
                                 ;-

```

```
                                                                F  4
EVTTBL     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00
SYMBOL  VALUE          REFERENCES

                        3-94     #3-95     3-95    #3-96     3-96    #3-97     3-97    #3-98     3-98
                       #3-98      3-98    #3-98     3-98    #3-98     3-98    #3-98     3-98    #3-99
                        3-99     #3-100    3-100   #3-101    3-101   #3-102    3-102   #3-103    3-103
                       #3-104     3-104   #3-105    3-105   #3-106    3-106   #3-107    3-107   #3-107
                        3-107    #3-107    3-107   #3-107    3-107   #3-107    3-107   #3-107    3-107
$$$W2    = 000000      #3-88      3-88    #3-89     3-89    #3-90     3-90    #3-91     3-91    #3-92
                        3-92     #3-93     3-93    #3-94     3-94    #3-95     3-95    #3-96     3-96
                       #3-97      3-97    #3-98     3-98    #3-99     3-99    #3-100    3-100   #3-101
                        3-101    #3-102    3-102   #3-103    3-103   #3-104    3-104   #3-105    3-105
                       #3-106     3-106   #3-107    3-107
$$$W3    = 000000      #3-88      3-88    #3-89     3-89    #3-90     3-90    #3-91     3-91    #3-92
                        3-92     #3-93     3-93    #3-94     3-94    #3-95     3-95    #3-96     3-96
                       #3-97      3-97    #3-98     3-98    #3-99     3-99    #3-100    3-100   #3-101
                        3-101    #3-102    3-102   #3-103    3-103   #3-104    3-104   #3-105    3-105
                       #3-106     3-106   #3-107    3-107
...N     = 000000      #3-50      3-54     3-54    #3-54     3-54     3-54    #3-54     3-54     3-54
                       #3-54      3-54     3-54    #3-54     3-54     3-54    #3-54     3-54     3-54
                       #3-54      3-54     3-54    #3-54     3-54     3-54    #3-54     3-54     3-54
                       #3-54      3-54     3-54    #3-54     3-54     3-54    #3-54     3-54     3-54
                       #3-54      3-54     3-54    #3-54     3-54     3-54    #3-54     3-54     3-54
                       #3-54      3-54     3-54    #3-54
```

```
                                                                G  4
EVTTBL     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 3
```

```
   187 000000                                     PRMPT$  <"Location in memory: ">,                P$LOC,   <:>
   188 000000                                     PRMPT$  <"Logging sink type: ">,                 P$LGID,  <:>
   189 000000                                     PRMPT$  <"Logging state: ">,                     P$LOST,  <:>
   190 000000                                     PRMPT$  <"Loop component: ">,                    P$LOO,   <:>
   191 000000                                     PRMPT$  <"Loop help type: ">,                    P$LHE,   <:>
   192 000000                                     PRMPT$  <"Maximum address: ">,                   P$MAD,   <:>
   193 000000                                     PRMPT$  <"Maximum Area Number: ">,               P$AREA,  <:>
   194 000000                                     PRMPT$  <"Maximum block (bytes): ">,             P$LMB,   <:>
   195 000000                                     PRMPT$  <"Maximum Broadcast Routers: ">,         P$MBR,   <:>
   196 000000                                     PRMPT$  <"Maximum Broadcast NonRouters: ">,      P$MBN,   <:>
   197 000000                                     PRMPT$  <"Maximum buffers: ">,                   P$MBU,   <:>
   198 000000                                     PRMPT$  <"Maximum circuits: ">,                  P$MLN,   <:>
   199 000000                                     PRMPT$  <"Maximum clears: ">,                    P$LMX,   <:>
   200 000000                                     PRMPT$  <"Maximum controllers: ">,               P$MXCO,  <:>
   201 000000                                     PRMPT$  <"Maximum cost: ">,                      P$MCO,   <:>
   202 000000                                     PRMPT$  <"Maximum data (bytes): ">,              P$CMB,   <:>
   203 000000                                     PRMPT$  <"Maximum hops: ">,                      P$MHO,   <:>
   204 000000                                     PRMPT$  <"Maximum lines: ">,                     P$MXLI,  <:>
   205 000000                                     PRMPT$  <"Maximum links: ">,                     P$MLK,   <:>
   206 000000                                     PRMPT$  <"Maximum recall count: ">,              P$MXR,   <:>
   207 000000                                     PRMPT$  <"Maximum receive buffers: ">,           P$MRB,   <:>
   208 000000                                     PRMPT$  <"Maximum resets: ">,                    P$RMX,   <:>
   209 000000                                     PRMPT$  <"Maximum restarts: ">,                  P$SMX,   <:>
   210 000000                                     PRMPT$  <"Maximum retransmits: ">,               P$MRT,   <:>
   211 000000                                     PRMPT$  <"Maximum Routers: ">,                   P$MXRT   <:>
   212 000000                                     PRMPT$  <"Maximum transmit count: ">,            P$MXB,   <:>
   213 000000                                     PRMPT$  <"Maximum visits: ">,                    P$MVI,   <:>
   214 000000                                     PRMPT$  <"Maximum window (blks): ">,             P$MXW,   <:>
   215 000000                                     PRMPT$  <"Module identification: ">,             P$MOID,  <:>
   216 000000                                     PRMPT$  <"Multipoint ratio type: ">,             P$MUL,   <:>
   217 000000                                     PRMPT$  <"Network name: ">,                      P$NET,   <:>
   218 000000                                     PRMPT$  <"Node identification: ">,               P$NID,   <:>
   219 000000                                     PRMPT$  <"Node name: ">,                         P$NAM,   <:>
   220 000000                                     PRMPT$  <"Executor state: ">,                    P$NOST,  <:>
   221 000000                                     PRMPT$  <"Executor Node Type: ">,                P$ETY,   <:>
   222 000000                                     PRMPT$  <"Number of test messages: ">,           P$COU,   <:>
   223 000000                                     PRMPT$  <"Object identification: ">,             P$OBJ,   <:>
   224 000000                                     PRMPT$  <"Object name: ">,                       P$OBNA,  <:>
   225 000000                                     PRMPT$  <"Object number: ">,                     P$OID,   <:>
   226 000000                                     PRMPT$  <"Object user: ">,                       P$OBUS,  <:>
   227 000000                                     PRMPT$  <"Object verification: ">,               P$OBVE,  <:>
   228 000000                                     PRMPT$  <"Originating Queue Limit: ">,           P$OQL,   <:>
   229 000000                                     PRMPT$  <"Outgoing timer (sec): ">,              P$OTI,   <:>
   230 000000                                     PRMPT$  <"Output file/device: ">,                P$TO,    <:>
   231 000000                                     PRMPT$  <"Partition name: ">,                    P$PAR,   <:>
   232 000000                                     PRMPT$  <"Password: ">,                          P$PAS,   <:>
   233 000000                                     PRMPT$  <"Physical address: ">                   P$PHA,   <:>
   234 000000                                     PRMPT$  <"Physical option: ">                    P$PHO,   <:>
   235 000000                                     PRMPT$  <"Polling state: ">,                     P$PST,   <:>
   236 000000                                     PRMPT$  <"Priority: ">,                          P$PRI,   <:>
   237 000000                                     PRMPT$  <"Process name: ">,                      P$PID,   <:>
   238 000000                                     PRMPT$  <"Router Priority: ">,                   P$RTPR   <:>
   239 000000                                     PRMPT$  <"Recall timer (sec): ">,                P$RET,   <:>
   240 000000                                     PRMPT$  <"Receive buffers: ">,                   P$BFQ,   <:>
   241 000000                                     PRMPT$  <"Reset timer (sec): ">,                 P$RST,   <:>
   242 000000                                     PRMPT$  <"Restart timer (sec): ">,               P$RRT,   <:>
   243 000000                                     PRMPT$  <"Retransmit factor: ">,                 P$RFA,   <:>
```

```
    244 000000                                  PRMPT$  <"Retransmit timer (msec): ">,          P$RTR,  <:>
    245 000000                                  PRMPT$  <"Routing timer (sec): ">,              P$RTI,  <:>
    246 000000                                  PRMPT$  <"Scheduling timer (msec): ">,          P$SCT,  <:>
    247 000000                                  PRMPT$  <"Scope: ">,                            P$SCO,  <:>
    248 000000                                  PRMPT$  <"Secondary dumper file: ">,            P$SDU,  <:>
    249 000000                                  PRMPT$  <"Secondary loader file: ">,            P$SLO,  <:>
    250 000000                                  PRMPT$  <"Secondary loader input file: ">,      P$SEC,  <:>
    251 000000                                  PRMPT$  <"Segment buffer size: ">,              P$SGZ,  <:>
    252 000000                                  PRMPT$  <"Service option: ">,                   P$SERO, <:>
    253 000000                                  PRMPT$  <"Service circuit id: ">,               P$SLI,  <:>
    254 000000                                  PRMPT$  <"Service device: ">,                   P$SDV,  <:>
    255 000000                                  PRMPT$  <"Service Node Version: ">,             P$SNV,  <:>
    256 000000                                  PRMPT$  <"Service state: ">,                    P$SER,  <:>
    257 000000                                  PRMPT$  <"Service timer (msec): ">,             P$STI,  <:>
    258 000000                                  PRMPT$  <"Set alias options: ">,                P$SEAL, <:>
    259 000000                                  PRMPT$  <"Set circuit options: ">,              P$SECI, <:>
    260 000000                                  PRMPT$  <"Set component: ">,                    P$SET,  <:>
    261 000000                                  PRMPT$  <"Set line options: ">,                 P$SELI, <:>
    262 000000                                  PRMPT$  <"Set logging options: ">,              P$SELO, <:>
    263 000000                                  PRMPT$  <"Set X25-access options: ">,           P$SEXA, <:>
    264 000000                                  PRMPT$  <"Set X25-protocol options: ">,         P$SEXP, <:>
    265 000000                                  PRMPT$  <"Set X25-server options: ">,           P$SE5S, <:>
    266 000000                                  PRMPT$  <"Set X29-server options: ">,           P$SE9S, <:>
    267 000000                                  PRMPT$  <"Set node options: ">,                 P$SENO, <:>
    268 000000                                  PRMPT$  <"Set object options: ">,               P$SEOB, <:>
    269 000000                                  PRMPT$  <"Set process options: ">,              P$SEPR, <:>
    270 000000                                  PRMPT$  <"Set trace options: ">,                P$SETR, <:>
    271 000000                                  PRMPT$  <"Show component: ">,                   P$SHO,  <:>
    272 000000                                  PRMPT$  <"Sink name: ">,                        P$SNAM, <:>
    273 000000                                  PRMPT$  <"Sink node identification: ">,         P$SIN,  <:>
    274 000000                                  PRMPT$  <"Software id: ">,                      P$SID,  <:>
    275 000000                                  PRMPT$  <"Software type: ">,                    P$STY,  <:>
    276 000000                                  PRMPT$  <"State: ">,                            P$STA,  <:>
    277 000000                                  PRMPT$  <"Stream timer (msec): ">,              P$STT,  <:>
    278 000000                                  PRMPT$  <"Subaddresses: ">,                     P$SUB,  <:>
    279 000000                                  PRMPT$  <"Software option: ">,                  P$SWO,  <:>
    280 000000                                  PRMPT$  <"Terminal identification: ">,         P$TER,  <:>
    281 000000                                  PRMPT$  <"Tertiary loader input file: ">,       P$TLO,  <:>
    282 000000                                  PRMPT$  <"Text password: ">,                    P$TPAS, <:>
    283 000000                                  PRMPT$  <"Trace file: ">,                       P$TFIL, <:>
    284 000000                                  PRMPT$  <"Trace state: ">,                      P$TSTA, <:>
    285 000000                                  PRMPT$  <"Transmit timer (msec): ">,            P$XMT,  <:>
    286 000000                                  PRMPT$  <"Tributary address: ">,                P$TRIB, <:>
    287 000000                                  PRMPT$  <"Trigger options: ">,                  P$TRIG, <:>
    288 000000                                  PRMPT$  <"CUG Type: ">,                         P$GTY,  <:>
    289 000000                                  PRMPT$  <"Type of test data: ">,                P$WIT,  <:>
    290 000000                                  PRMPT$  <"Usage type: ">,                       P$CUS,  <:>
    291 000000                                  PRMPT$  <"User identification: ">,              P$USE,  <:>
    292 000000                                  PRMPT$  <"Vector: ">,                           P$VEC,  <:>
    293 000000                                  PRMPT$  <"Verification state: ">,               P$VEST, <:>
    294 000000                                  PRMPT$  <"Zero component: ">,                   P$ZER,  <:>
    295
    296          000001                         .END
```

```
                                                .SBTTL  .EXNAM - EXECUTOR NODE NAME ACTION ROUTINE
                                        ;+
                                        ; *** - .EXNAM - EXECUTOR NODE NAME ACTION ROUTINE
                                        ;
                                        ; INPUT:
                                        ;       .PSTPT = ADDRESS OF NODE NAME
                                        ;       .PSTCN = COUNT OF CHARS IN NODE NAME
                                        ;
                                        ; OUTPUT:
                                        ;       $NODNA = RAD50 NODE NAME (2 WORDS)
                                        ;       R0,R1 = DESTROYED
                                        ;       THE TRANSITION IS ACCEPTED OR REJECTED
                                        ;
                                        ;-

000344                                  .EXNAM::
000344  004767  000024                          CALL    NODNAM          ; CONVERT NODE NAME
000350  103406                                  BCS     10$             ; IF CS, ERROR
000352  004767  000676                          CALL    FNDNAM          ; ELSE, TRY TO FIND NODE NAME IN REMOTES
000356  103405                                  BCS     20$             ; IF CS, CAN'T .. OKAY
000360  012767  000000' 000000G                 MOV     #ERRNAM,$ERROR  ; ELSE, NAME ALREADY IN USE
000366  062716  000002                  10$:    ADD     #2,(SP)         ; REJECT TRANSITION
000372  000207                          20$:    RETURN
```

```
PRSACT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:17      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE             REFERENCES

.RNAD2     000460 RG       #11-256
.RNAD3     000650 RG       #11-318
.RNHNA     001172 RG       #15-438
.RNINI     000452 RG       #11-240
.RNNAM     001052 RG       #14-400
```

```
**FILE**ID**PRSCIR
```

```
                                        .SBTTL  CAT5 - CONVERT ASCII TO RAD50
                              ;+
                              ; **-CAT5-CONVERT ASCII TO RAD50
                              ;
                              ; INPUTS:
                              ;
                              ;         R0 = ADDRESS OF FIRST BYTE IN STRING
                              ;         R3 = LEGAL TERMINATING CHARACTER
                              ;         R4 = ADDRESS OF THE END OF THE STRING + 1
                              ;
                              ;
                              ; OUTPUTS:
                              ;
                              ;         C-BIT CLEAR:
                              ;         R0 = ADDRESS OF NEXT BYTE IN STRING
                              ;         R1 = CONVERTED RESULT
                              ;         R2 = COMPLETION CODE (0 = MORE IN THE STRING)
                              ;                              (NOT 0 = NO MORE IN THE STRING)
                              ;
                              ;         C-BIT SET:
                              ;         ILLEGAL CHARACTER IN THE STRING
                              ;
                              ; REGISTERS MODIFIED:
                              ;
                              ;         NONE
                              ;-

000556  003100                CVTBL:  .WORD   50*50
000560  000050                        .WORD   50
000562  000001                        .WORD   1

000564  010546                CAT5:   MOV     R5,-(SP)          ; SAVE REGISTERS
000566  005046                        CLR     -(SP)             ; ESTABLISH EXIT CONDITION FLAG
000570  005001                        CLR     R1                ; CLEAR ACCUMULATION
000572  012702  000556'               MOV     #CVTBL,R2         ; SET ADDRESS OF SCALING TABLE
000576  112005                10$:    MOVB    (R0)+,R5          ; GET A CHARACTER
000600  020400                        CMP     R4,R0             ; REACHED THE END OF THE STRING ?
000602  103434                        BLO     30$               ; IF LO, YES - END OF THE STRING
000604  122705  000132                CMPB    #'Z,R5            ; IS IT ALPHABETIC ? (BELOW Z)
000610  103425                        BLO     20$               ; IF LO, NO - ERROR
000612  122705  000101                CMPB    #'A,R5            ; IS IT BELOW A ?
000616  101022                        BHI     20$               ; IF HI, YES - ERROR
000620  162705  000100                SUB     #100,R5           ; SCALE TO RAD50
000624  010046                        MOV     R0,-(SP)          ; SAVE R0
000626  012200                        MOV     (R2)+,R0          ; GET SCALE FACTOR
000630  010146                        MOV     R1,-(SP)          ; SAVE R1
000632  010501                        MOV     R5,R1             ; COPY CHARACTER
000634                                CALL    $MUL              ; MULTIPLY IT OUT
000640  010105                        MOV     R1,R5             ; COPY RESULT
000642  012601                        MOV     (SP)+,R1          ; RESTORE ACCUMULATION
000644  012600                        MOV     (SP)+,R0          ; RESTORE STRING POINTER
000646  060501                        ADD     R5,R1             ; ACCUMULATE THE RESULT
000650  020227  000564'               CMP     R2,#CVTBL+6       ; CONVERTED ENOUGH ?
000654  103750                        BLO     10$               ; IF LO, NO - KEEP GOING
000656  112005                        MOVB    (R0)+,R5          ; OTHERWISE GET TERMINATING CHARACTER
000660  020400                        CMP     R4,R0             ; REACHED THE END OF THE STRING ?
000662  101404                        BLOS    30$               ; IF LO, YES.  (if eq, premature end)        ;04
```

```
                                                                        F 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

BIT0     = 000001           #6-74
BIT1     = 000002           #6-74
BIT10    = 002000           #6-74
BIT11    = 004000           #6-74
BIT12    = 010000           #6-74
BIT13    = 020000           #6-74
BIT14    = 040000           #6-74
BIT15    = 100000           #6-74
BIT2     = 000004           #6-74
BIT3     = 000010           #6-74
BIT4     = 000020           #6-74
BIT5     = 000040           #6-74
BIT6     = 000100           #6-74
BIT7     = 000200           #6-74
BIT8     = 000400           #6-74
BIT9     = 001000           #6-74
B$10       000003           #5-74
B$11       000004           #5-74
B$12       000005           #5-74
B$5        000000           #5-74
B$6        000001           #5-74
B$7        000002           #5-74
B$7S10     000006           #5-74
CAT5       000564 R          8-171       #11-365
CDTB       000706 R          8-189        8-208         8-234         #12-430
CHKDEV     001036 R          8-243       #14-506
CHKMUX     001004 R          8-179       #13-475
CHKWLD     001372 R          8-183        8-200         8-226         #16-663
CVTBL      000556 R         #11-361       11-368        11-386
HF$GWY   = 000010            10-312
H$FLG      000000            10-312
I$$AS    = ******            5-64
LF$END   = 000020           #6-74
LF$MLT   = 040000           #6-74
LF$REA   = 000001           #6-74
LF$SEG   = 100000           #6-74
LF$SIG   = 000040           #6-74
LF$SKP   = 000004           #6-74
LF$VR2   = 000010           #6-74
LF$ZER   = 000002           #6-74
LF.ACT   = 100000           #5-65
LF.BRO   = 000400           #5-65
LF.BWT   = 000007           #5-65
LF.ENA   = 002000           #5-65
LF.LPB   = 001000           #5-65
LF.MDC   = 000100           #5-65
LF.MFL   = 004000           #5-65
LF.MTP   = 000020           #5-65         14-573        14-593
LF.PAC   = 000200           #5-65
LF.RDY   = 040000           #5-65
LF.REA   = 010000           #5-65
LF.SER   = 000040           #5-65
```

```
                                                                        G 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 2
```

```
                                                .SBTTL  .EXNN - SET MAXIMUM NODE NUMBER IN NETWORK
                                ;+
                                ; *** - .EXNN - SET MAXIMUM NODE NUMBER IN NETWORK
                                ;
                                ; INPUTS:
                                ;       .PNUMB = THE MAXIMUM NODE NUMBER IN NETWORK
                                ;
                                ; OUTPUTS:
                                ;       $NN = THE MAXIMUM NODE NUMBER IN NETWORK
                                ;       THE  TRANSITION IS ACCEPTED/REJECTED.
                                ;
                                ;-

000274                          .EXNN::
000274  012700  000000G                 MOV     #$TMLST,R0        ; SCAN NODE LIST FOR ROUTING TEMPLATE
000300  011000                  1$:     MOV     (R0),R0           ; NEXT TEMPLATE
000302  001414                          BEQ     5$                ; IF EQ, NO ROUTING TEMPLATE
000304  122760  000000G 000000G         CMPB    #CS.ROU,C.STS(R0) ; IS THIS IT ?
000312  001372                          BNE     1$                ; IF NE, NOPE
000314  005760  000004                  TST     4(R0)             ; GOT IT -TEST FOR END NODE
000320  001405                          BEQ     5$                ; IF EQ, END NODE
000322  016767  000000G 000000G         MOV     .PNUMB,$NN        ; STORE THE MAX. NODE NUMBER
000330  001405                          BEQ     10$               ; IF EQ, ILLEGAL VALUE
                                ;;;     CMP     .PNUMB,#MAXNN     ; IS THE VALUE IN RANGE ?
                                ;;;     BHI     10$               ; IF HI, OUT OF RANGE
000332  000411                          BR      20$               ; ELSE OKAY

000334  012767  000000G 000000G 5$:     MOV     #ERRPNA,$ERROR    ; PARAMETER NOT APPLICABLE
000342  000403                          BR      15$               ; JOIN COMMON CODE

000344  012767  000000G 000000G 10$:    MOV     #ERRILN,$ERROR    ; INVALID NUMBER
000352  062716  000002          15$:    ADD     #2,(SP)           ; ELSE, REJECT TRANSITION

000356  000207                  20$:    RETURN
```

```
                                                             F 12
PRSEXE    CREATED BY  MACRO  ON 29-JUN-85 AT 05:18    PAGE 1
SYMBOL CROSS REFERENCE                                CREF   04.00

SYMBOL  VALUE               REFERENCES

CS.ROU  = ******  GX     9-207    10-248
C.ADD   = 000004         #4-67    10-250
C.STS   = ******  GX     9-207    10-248
ERRILN  = ******  GX     6-143    7-165      8-187      9-220      10-256     12-319     13-366     14-388     15-409
                        16-428    17-449     18-470     19-491     20-512
ERRPNA  = ******  GX     9-217    13-363
MAXCST  = ******  GX     8-184    13-349
MAXHOP  = ******  GX     7-162    13-356
MAXNN   = ******  GX    10-252
NF.LV2  = ******  GX    13-361
OE.NN   = ******  GX    10-243
STRMX   = 000040         #4-66    5-110      5-116
$CNFIG  = ******  GX    13-361
$EACOS  = ******  GX   *13-347
$EAHOP  = ******  GX   *13-354
$EAMAX  = ******  GX   *13-340
$EBEA   = ******  GX   *15-402
$EBRT   = ******  GX   *14-383
$EINA   = ******  GX   *17-442
$EINC   = ******  GX   *18-463
$EOUT   = ******  GX   *19-484
$ERET   = ******  GX   *20-505
$ERROR  = ******  GX    *6-143    *7-165     *8-187     *9-217     *9-220     *10-256    *12-319    *13-363    *13-366
                       *14-388    *15-409    *16-428    *17-449    *18-470    *19-491    *20-512
$ESEG   = ******  GX   *16-423
$ESUB   = ******  GX    12-306    12-310     *12-311
$HOST   = ******  GX   *11-272    11-277     *11-278     11-282    *11-285
$ID     = ******  GX     5-109    5-120
$LINKS  = ******  GX    *6-138
$MAXC   = ******  GX    *8-182
$MAXH   = ******  GX    *7-160
$MVASC  = ******  GX     5-102    5-122
$NCOU   = ******  GX   *10-239
$NN     = ******  GX    *9-211    10-242
$NODAD  = ******  GX    11-272    11-276     *11-277
$NTADD  = ******  GX    11-280    11-285
$OPRND  = ******  GX    10-243
$TMLST  = ******  GX     9-204    10-245
.EACOS    000644 RG    #13-346
.EAHOP    000666 RG    #13-353
.EAMAX    000624 RG    #13-339
.EXBEA    001002 RG    #15-401
.EXBRT    000750 RG    #14-382
.EXCOS    000240 RG     #8-181
.EXHOP    000204 RG     #7-159
.EXHOS    000476 RG    #11-275
.EXHO1    000466 RG    #11-271
.EXID     000000 RG      #5-85
.EXID1    000064 RG       5-92    #5-107
.EXINA    001076 RG    #17-441
.EXINT    001136 RG    #18-462
.EXLNK    000152 RG     #6-137
```

```
                                                             G 12
PRSEXE    CREATED BY  MACRO  ON 29-JUN-85 AT 05:18    PAGE 2
```

```
                                        .SBTTL  CDTB - CONVERT DECIMAL TO BINARY
                                ;+
                                ; **-CDTB-CONVERT DECIMAL TO BINARY
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = ADDRESS OF THE FIRST CHARACTER TO CONVERT
                                ;       R3 = LEGAL TERMINATING CHARACTER
                                ;       R4 = ADDRESS OF THE END OF THE STRING + 1
                                ;
                                ; OUTPUTS:
                                ;
                                ;       C-BIT CLEAR:
                                ;       R0 = ADDRESS OF THE NEXT CHARACTER IN THE STRING
                                ;       R1 = CONVERTED RESULT
                                ;       R2 = COMPLETION CODE (0 = MORE IN THE STRING)
                                ;                            (NOT 0 = END OF THE STRING)
                                ;
                                ;       C-BIT SET:
                                ;       ILLEGAL CHARACTER IN THE STRING
                                ;
                                ; REGISTERS MODIFIED:
                                ;
                                ;       NONE
                                ;-

000400  005046                  CDTB:   CLR     -(SP)                   ; SET COMPLETION FLAGS
000402  005001                          CLR     R1                      ; CLEAR ACCUMULATION
000404  112002                  10$:    MOVB    (R0)+,R2                ; GET THE NEXT CHARACTER
000406  020400                          CMP     R4,R0                   ; REACHED THE END OF THE STRING ?
000410  103421                          BLO     20$                     ; IF LO, YES - EXIT WITH RESULT
000412  120302                          CMPB    R3,R2                   ; IS THE CHARACTER A LEGAL TERMINATOR ?
000414  001415                          BEQ     15$                     ; IF EQ, YES - EXIT WITH RESULT        ;01
000416  162702  000060                  SUB     #'0,R2                  ; NORMALIZE THE CHARACTER
000422  020227  000012                  CMP     R2,#10.                 ; IS IT WITHIN RANGE ?
000426  103016                          BHIS    40$                     ; IF HIS, NO - EXIT WITH ERROR
000430  010046                          MOV     R0,-(SP)                ; SAVE THE STRING POINTER
000432  012700  000012                  MOV     #10.,R0                 ; SET THE MULTIPLICAND
000436                                  CALL    $MUL                    ; MULTIPLY 'EM UP
000442  012600                          MOV     (SP)+,R0                ; RESTORE THE STRING POINTER
000444  060201                          ADD     R2,R1                   ; ACCUMULATE THE RESULT
000446  000756                          BR      10$                     ; CONTINUE

000450  020400                  15$:    CMP     R4,R0                   ; WAS 'LEGAL TERMINATOR' THE END-OF-STRING ?;01
000452  001001                          BNE     30$                     ; IF NE, NO. CONTINUE                   ;01

000454  010616                  20$:    MOV     SP,(SP)                 ; SET THE STRING END INDICATOR

000456  022701  000400          30$:    CMP     #256.,R1                ; IS THE RESULT WITH A BYTE QUANTITY ?
000462  101002                          BHI     50$                     ; IF HI, YES - EXIT WITH RESULT
000464  000261                  40$:    SEC                             ; INDICATE ILLEGAL CHARACTER IN STRING
000466  000401                          BR      60$                     ;   OR RESULT OUT OF RANGE

000470  000241                  50$:    CLC                             ; INDICATE SUCCESSFUL CONVERSION
000472  012602                  60$:    MOV     (SP)+,R2                ; SET COMPLETION CONDITIONS
000474                                  RETURN                          ; RETURN TO CALLER
```

```
                                                                F 14
PRSLIN      CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 4
SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE              REFERENCES

S.LEN      000004          #5-60
S.NMST     000002          #5-60
S.OWNR     000003          #5-60
X$$MCB  =  ******            5-59       5-59
ZF.COU  =  001000          #5-59
ZF.DDM  =  000001          #5-59        12-417
ZF.DIA  =  004000          #5-59
ZF.DLC  =  000002          #5-59
ZF.DVP  =  100000          #5-59
ZF.INI  =  040000          #5-59
ZF.KMX  =  000020          #5-59
ZF.LLC  =  000004          #5-59
ZF.LMC  =  000100          #5-59
ZF.MAN  =  020000          #5-59
ZF.MFL  =  000010          #5-59
ZF.MTM  =  000400          #5-59
ZF.MUX  =  000040          #5-59        10-313
ZF.PSE  =  002000          #5-59        12-415
ZF.SLI  =  010000          #5-59
ZF.TIM  =  000200          #5-59
ZF.X3P  =  000000          #5-59
ZS.ASN  =  100000          #5-59
ZS.BSY  =  140000          #5-59
Z.AVL      000014          #5-59
Z.DAT      000016          #5-59
Z.DSP      000000          #5-59        5-59
Z.FLG      000010          #5-59        10-313      12-415      12-417
Z.LEN   =  000016          #5-59
Z.LLN      000006          #5-59
Z.MAP      000020          #5-59
Z.NAM      000004          #5-59
Z.PCB      000012          #5-59
Z.SCH      000007          #5-59
$HEADR  =  ******    GX     11-387
$MUL    =  ******    GX     8-215       9-276
$SAVAL  =  *** **    GX     7-91
```

```
                                                                G 14
PRSLIN      CREATED BY  MACRO  ON 29-JUN-85 AT 13:23      PAGE 5
```

F 15

```
PRSLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE              REFERENCES

CKEVE     000000 R         #5-90        8-215       9-252
CKEVEB    000064 RG        #6-138
CKEVEC    000114 RG        #7-172
CKEVEN    000204 RG        #8-213
CKEVER    000224 RG        #9-244
SPRAY     000270 RG         7-186      #10-281
$CLASS  = ******  GX       *7-182
$EVBEG  = ******  GX       *6-143       9-245
$EVENT  = ******  GX       *5-107       7-184
.PNUMB  = ******  GX        6-139       7-175       8-214       9-248       9-250
.PNUMH  = ******  GX        6-141       7-173       9-246
```

G 15

```
PRSLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 2
```

```
                                         .SBTTL  .MORET - PARSE RESET TIMER
                                ;+
                                ; ** .MORET - PARSE RESET TIMER
                                ;
                                ; INPUTS         .TPARS VARIABLES.
                                ;
                                ; OUTPUT         $RET = PARSED RESET TIMER
                                ;                THE TRANSITION IS ACCEPTED/REJECTED.
                                ;
                                ;-
000324                          .MORET::
000324  005767  000000G                  TST     .PNUMH          ; HIGH ORDER WORD
000330  001066                           BNE     ILLNUM          ; REJECT IF TOO HIGH
000332  105767  000001G                  TSTB    .PNUMB+1        ; TEST HIGH BYTE
000336  001063                           BNE     ILLNUM          ; BR IF  ILLEGAL NUMBER
000340  116767  000000G 000000G          MOVB    .PNUMB,$RET     ; MOVE VALUE TO PARSED AREA
000346  000207                           RETURN
```

```
Symbol table

BYTFLG  000006R         $GTNMB  000010RG        .CAT5 = ****** GX       .PNUMB= ****** GX       .PSTPT= ****** GX
REJECT  000200R         $GTNUM  000016RG        .ERROR  000000RG        .PNUMH= ****** GX       .TEMP   000004RG
$CLERR  000206RG        $GTR50  000072RG        .FLAGS  000002RG        .PSTCN= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000214    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 46 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.11
SY:CFETSB.V2,[132,134]CFETSB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFETSB
```

```
                                   ;+
                                   ; *** - $MOVPT - MOVE THE CURRENT TPARS FIELD
                                   ;
                                   ; INPUT:
                                   ;         .PSTPT = ADDRESS OF CURRENT FIELD
                                   ;         .PSTCN = LENGTH OF CURRENT FIELD
                                   ;         $OBUF = CURRENT POSITION IN THE OUTPUT BUFFER
                                   ;
                                   ; OUTPUT:
                                   ;         $OBUF = NEXT LOCATION IN THE OUTPUT BUFFER
                                   ;
                                   ;-

 000476                            $MOVPT::
 000476  016702  000000G                   MOV     .PSTCN,R2       ; GET THE LENGTH OF CURRENT FIELD
 000502  001410                            BEQ     10$             ; IF ZERO, NOTHING TO MOVE
 000504  016701  000000G                   MOV     .PSTPT,R1       ; POINT R1 AT INPUT STRING
 000510  016700  177270                    MOV     $OBUF,R0        ; POINT R0 AT OUTPUT BUFFER
 000514  004767  000000G                   CALL    $MVASC          ; MOVE THE STRING INTO THE OUTPUT BUFFER
 000520  010067  177260                    MOV     R0,$OBUF        ; SAVE NEXT LOCATION IN OUTPUT BUFFER
 000524  000207                    10$:    RETURN                  ; RETURN TO CALLER
```

```
                                                                   G  3
CFEUSB - TEMPLATE UPDATE SUBROU MACRO V05.03b Saturday 29-Jun-85 05:16  Page 5-1
$PTR50 - PUT RAD50

    133 000224                          $PTR50::
    134 000224  004767  000064                   CALL     CHKOPT          ; OUTPUT COMMAS FOR ANY MISSING PARAMS
    135 000230  010346                           MOV      R3,-(SP)        ; FREE UP SOME REGISTERS
    136 000232  010446                           MOV      R4,-(SP)        ;  ...
    137 000234  010103                           MOV      R1,R3           ; COPY WORD COUNT
    138 000236  010204                           MOV      R2,R4           ; COPY OFFSET
    139 000240  066704  177540                   ADD      .TEMP,R4        ; POINT TO FIELD
    140 000244  016700  000000G                  MOV      $OBUF,R0        ; POINT TO OUTPUT BUFFER
    141 000250  012401                  10$:     MOV      (R4)+,R1        ; GET A TEMPLATE WORD
    142 000252  004767  000000G                  CALL     $C5TA           ; CONVERT RAD50 TO ASCII
    143 000256  005303                           DEC      R3              ; DECREASE WORD COUNT
    144 000260  003373                           BGT      10$             ; IF GT, MORE
    145 000262  010067  000000G                  MOV      R0,$OBUF        ; RESTORE BUFFER POINTER
    146 000266  012604                           MOV      (SP)+,R4        ; RESTORE REGISTERS
    147 000270  012603                           MOV      (SP)+,R3        ;  ...
    148 000272  000207                           RETURN
    149
```

```
                                                                   H  3
CFEUSB - TEMPLATE UPDATE SUBROU MACRO V05.03b Saturday 29-Jun-85 05:16  Page 6
$PTOPT - CHECK FOR OPTIONAL PARAMETER
```

```
                                                                      G  4
EVTTBL     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME     REFERENCES

EVENT          #3-60     3-88      3-89      3-90      3-91      3-92      3-93      3-94      3-95      3-96
                3-97     3-98      3-99      3-100     3-101     3-102     3-103     3-104     3-105     3-106
                3-107
...BIS         #3-56     3-88      3-88      3-88      3-88      3-88      3-88      3-88      3-88      3-88
                3-88     3-89      3-89      3-90      3-90      3-90      3-91      3-91      3-91      3-91
                3-91     3-91      3-91      3-91      3-91      3-91      3-91      3-91      3-91      3-91
                3-91     3-91      3-91      3-91      3-91      3-91      3-92      3-92      3-92      3-92
                3-92     3-92      3-92      3-92      3-92      3-92      3-92      3-92      3-92      3-92
                3-92     3-92      3-92      3-92      3-92      3-93      3-93      3-93      3-93      3-93
                3-93     3-94      3-94      3-94      3-95      3-96      3-97      3-98      3-98      3-98
                3-98     3-98      3-99      3-100     3-101     3-101     3-101     3-101     3-102     3-102
                3-102    3-103     3-104     3-105     3-106     3-106     3-106     3-106     3-106     3-106
                3-106    3-106     3-106     3-106     3-106     3-107     3-107     3-107     3-107     3-107
                3-107    3-107     3-107     3-107     3-107     3-107     3-107     3-107     3-107     3-107
                3-107    3-107     3-107     3-107     3-107     3-107
```

```
                                                                      H  4
**FILE**ID**NOSNA
```

```
   244 000000                    PRMPT$  <"Retransmit timer (msec): ">,        P$RTR,  <:>
   245 000000                    PRMPT$  <"Routing timer (sec): ">,            P$RTI,  <:>
   246 000000                    PRMPT$  <"Scheduling timer (msec): ">,        P$SCT,  <:>
   247 000000                    PRMPT$  <"Scope: ">,                          P$SCO,  <:>
   248 000000                    PRMPT$  <"Secondary dumper file: ">,          P$SDU,  <:>
   249 000000                    PRMPT$  <"Secondary loader file: ">,          P$SLO,  <:>
   250 000000                    PRMPT$  <"Secondary loader input file: ">,    P$SEC,  <:>
   251 000000                    PRMPT$  <"Segment buffer size: ">,            P$SGZ,  <:>
   252 000000                    PRMPT$  <"Service option: ">,                 P$SERO, <:>
   253 000000                    PRMPT$  <"Service circuit id: ">,             P$SLI,  <:>
   254 000000                    PRMPT$  <"Service device: ">,                 P$SDV,  <:>
   255 000000                    PRMPT$  <"Service Node Version: ">,           P$SNV,  <:>
   256 000000                    PRMPT$  <"Service state: ">,                  P$SER,  <:>
   257 000000                    PRMPT$  <"Service timer (msec): ">,           P$STI,  <:>
   258 000000                    PRMPT$  <"Set alias options: ">,              P$SEAL, <:>
   259 000000                    PRMPT$  <"Set circuit options: ">,            P$SECI, <:>
   260 000000                    PRMPT$  <"Set component: ">,                  P$SET,  <:>
   261 000000                    PRMPT$  <"Set line options: ">,               P$SELI, <:>
   262 000000                    PRMPT$  <"Set logging options: ">,            P$SELO, <:>
   263 000000                    PRMPT$  <"Set X25-access options: ">,         P$SEXA, <:>
   264 000000                    PRMPT$  <"Set X25-protocol options: ">,       P$SEXP, <:>
   265 000000                    PRMPT$  <"Set X25-server options: ">,         P$SE5S, <:>
   266 000000                    PRMPT$  <"Set X29-server options: ">,         P$SE9S, <:>
   267 000000                    PRMPT$  <"Set node options: ">,               P$SENO, <:>
   268 000000                    PRMPT$  <"Set object options: ">,             P$SEOB, <:>
   269 000000                    PRMPT$  <"Set process options: ">,            P$SEPR, <:>
   270 000000                    PRMPT$  <"Set trace options: ">,              P$SETR, <:>
   271 000000                    PRMPT$  <"Show component: ">,                 P$SHO,  <:>
   272 000000                    PRMPT$  <"Sink name: ">,                      P$SNAM, <:>
   273 000000                    PRMPT$  <"Sink node identification: ">,       P$SIN,  <:>
   274 000000                    PRMPT$  <"Software id: ">,                    P$SID,  <:>
   275 000000                    PRMPT$  <"Software type: ">,                  P$STY,  <:>
   276 000000                    PRMPT$  <"State: ">,                          P$STA,  <:>
   277 000000                    PRMPT$  <"Stream timer (msec): ">,            P$STT,  <:>
   278 000000                    PRMPT$  <"Subaddresses: ">,                   P$SUB,  <:>
   279 000000                    PRMPT$  <"Software option: ">,                P$SWO,  <:>
   280 000000                    PRMPT$  <"Terminal identification: ">,        P$TER,  <:>
   281 000000                    PRMPT$  <"Tertiary loader input file: ">,     P$TLO,  <:>
   282 000000                    PRMPT$  <"Text password: ">,                  P$TPAS, <:>
   283 000000                    PRMPT$  <"Trace file: ">,                     P$TFIL, <:>
   284 000000                    PRMPT$  <"Trace state: ">,                    P$TSTA, <:>
   285 000000                    PRMPT$  <"Transmit timer (msec): ">,          P$XMT,  <:>
   286 000000                    PRMPT$  <"Tributary address: ">,              P$TRIB, <:>
   287 000000                    PRMPT$  <"Trigger options: ">,                P$TRIG, <:>
   288 000000                    PRMPT$  <"CUG Type: ">,                       P$GTY,  <:>
   289 000000                    PRMPT$  <"Type of test data: ">,              P$WIT,  <:>
   290 000000                    PRMPT$  <"Usage type: ">,                     P$CUS,  <:>
   291 000000                    PRMPT$  <"User identification: ">,            P$USE,  <:>
   292 000000                    PRMPT$  <"Vector: ">,                         P$VEC,  <:>
   293 000000                    PRMPT$  <"Verification state: ">,             P$VEST, <:>
   294 000000                    PRMPT$  <"Zero component: ">,                 P$ZER,  <:>
   295
   296        000001             .END
```

```
                                                               G 6
PRMPTO - PROMPTS, OFFSET ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:02  Page 6-4
Symbol table

P$ACC   000010 G      P$CPU   001664 G      P$LHE   003700 G      P$PCH   000644 G      P$SNV   006526 G
P$ACT   000072 G      P$CSR   001702 G      P$LIST  003460 G      P$PEN   001536 G      P$STA   007472 G
P$ADD   000124 G      P$CTM   001634 G      P$LMB   004000 G      P$PHA   005560 G      P$STI   006602 G
P$AID   000140 G      P$CUS   010130 G      P$LMX   004204 G      P$PHO   005606 G      P$STT   007504 G
P$AMC   000210 G      P$CVL   000600 G      P$LNID  003366 G      P$PID   005672 G      P$STY   007450 G
P$AMH   000236 G      P$DDT   001776 G      P$LOA   003500 G      P$PRI   005654 G      P$SUB   007534 G
P$AREA  003750 G      P$DDTH  001752 G      P$LOC   003554 G      P$PRO   003436 G      P$SWO   007556 G
P$ARID  000160 G      P$DEA   001722 G      P$LOO   003654 G      P$PST   005632 G      P$TER   007602 G
P$AST   000312 G      P$DES   002230 G      P$LOST  003632 G      P$RET   005740 G      P$TFIL  007720 G
P$ASTN  000352 G      P$DEV   002256 G      P$LOWN  003416 G      P$RFA   006072 G      P$TLO   007636 G
P$ASTO  000264 G      P$DFA   002112 G      P$MAD   003724 G      P$RMX   004510 G      P$TO    005466 G
P$BBT   000376 G      P$DIA   002276 G      P$MBN   004070 G      P$RRT   006042 G      P$TPAS  007676 G
P$BDF   002024 G      P$DLT   002134 G      P$MBR   004032 G      P$RST   006014 G      P$TRIB  010012 G
P$BFQ   005770 G      P$DST   002206 G      P$MBU   004132 G      P$RTI   006154 G      P$TRIG  010040 G
P$BLK   000426 G      P$DTE   002322 G      P$MCO   004260 G      P$RTPR  005714 G      P$TSTA  007740 G
P$BRT   000452 G      P$DUA   002362 G      P$MHO   004334 G      P$RTR   006120 G      P$TYP   000766 G
P$BSA   000024 G      P$DUC   002404 G      P$MLK   004400 G      P$SCO   006240 G      P$USE   010150 G
P$BSD   002476 G      P$DUM   002424 G      P$MLN   004156 G      P$SCT   006204 G      P$VEC   010200 G
P$BSI   003146 G      P$DUMO  002442 G      P$MOID  004750 G      P$SDU   006252 G      P$VEST  010214 G
P$BUS   000514 G      P$DUP   002462 G      P$MRB   004454 G      P$SDV   006502 G      P$WDF   002056 G
P$CAT   000552 G      P$DWE   002164 G      P$MRT   004562 G      P$SEAL  006634 G      P$WIT   010102 G
P$CHN   000620 G      P$DYTH  002542 G      P$MUL   005002 G      P$SEC   006336 G      P$XMT   007760 G
P$CIID  000666 G      P$E1Y   005150 G      P$MVI   004672 G      P$SECI  006662 G      P$ZER   010244 G
P$CIST  000744 G      P$EVE   002566 G      P$MXB   004636 G      P$SELI  006734 G      X       000000RG      002
P$CLCI  001010 G      P$FNAM  002620 G      P$MXCO  004230 G      P$SELO  006762 G      Y       000000RG      003
P$CLE   001042 G      P$GNM   002656 G      P$MXLI  004356 G      P$SENO  007174 G      $ALPHA= 000022
P$CLK   001502 G      P$GRP   002636 G      P$MXR   004422 G      P$SEOB  007222 G      $ANY  = 000020
P$CLLI  001066 G      P$GTY   010064 G      P$MXRT  004612 G      P$SEPR  007252 G      $BLANK= 000006
P$CLLO  001116 G      P$HAD   002726 G      P$MXW   004716 G      P$SER   006560 G      $DIGIT= 000024
P$CLNO  001342 G      P$HAO   002754 G      P$NAM   005106 G      P$SERO  006426 G      $DNUMB= 000014
P$CLOB  001372 G      P$HBT   003032 G      P$NCP   000000 G      P$SET   006712 G      $EOS  = 000012
P$CLT   001424 G      P$HOS   003064 G      P$NET   005034 G      P$SETR  007302 G      $EXIT = 000000
P$CLTR  001452 G      P$HPAS  003000 G      P$NID   005056 G      P$SEXA  007012 G      $FAIL = 177777
P$CLXA  001150 G      P$HTM   002700 G      P$NOST  005124 G      P$SEXP  007046 G      $GOFF = 000001
P$CLXP  001206 G      P$IAT   003250 G      P$NUM   002342 G      P$SE5S  007104 G      $GPRM = 000001
P$CL5S  001246 G      P$IDE   003122 G      P$OBJ   005234 G      P$SE9S  007140 G      $LAMDA= 000000
P$CL9S  001304 G      P$INA   000044 G      P$OBNA  005266 G      P$SGZ   006376 G      $NUMBR= 000002
P$CMB   004302 G      P$IND   002516 G      P$OBUS  005330 G      P$SHO   007330 G      $RAD50= 000016
P$CMK   000534 G      P$INI   003170 G      P$OBVE  005350 G      P$SID   007430 G      $RONLY= 000001
P$COM   001522 G      P$INTH  003220 G      P$OID   005306 G      P$SIN   007372 G      $STRNG= 000004
P$CON   001562 G      P$ITI   003304 G      P$OQL   005400 G      P$SLI   006452 G      $SUBXP= 000010
P$COP   001606 G      P$LDOP  003532 G      P$OTI   005434 G      P$SLO   006304 G      $$$FLG= 177777
P$COS   001622 G      P$LEN   003336 G      P$PAR   005516 G      P$SMX   004534 G      $$$KEY= 177777
P$COU   005200 G      P$LGID  003604 G      P$PAS   005542 G      P$SNAM  007354 G      $$$LEN= 000022
P$COWN  000722 G

. ABS.  010270    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
$STATE  000000    002   (RO,D,LCL,REL,CON)
$KTAB   000000    003   (RO,D,LCL,REL,CON)
$KSTR   000000    004   (RO,D,LCL,REL,CON)
$PRMPT  000000    005   (RO,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics
```

```
                                                               H 6
PRMPTO - PROMPTS, OFFSET ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:02  Page 6-5
```

```
                                                .SBTTL  NODNAM - CONVERT NODE NAME TO RAD50
                                        ;+
                                        ; *** - NODNAM - CONVERT NODE NAME TO RAD50
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT = ADDRESS OF NODE NAME STRING
                                        ;       .PSTCN = LENGTH OF NODE NAME STRING
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       $NODNA = 2 WORD RAD50 NODE NAME
                                        ;
                                        ;
                                        ;-

 000374                                 NODNAM:
 000374 005067  000002G                         CLR     $NODNA+2        ; CLEAR THE SECOND HALF OF THE NODE NAME
 000400 016700  000000G                         MOV     .PSTPT,R0       ; POINT R0 AT NODE NAME
 000404 005001                                  CLR     R1              ; PERIODS NOT ACCEPTABLE
 000406 004767  000000G                         CALL    $CAT5           ; CONVERT NODE NAME TO RAD50
 000412 010167  000000G                         MOV     R1,$NODNA       ; STORE THE NODE NAME
 000416 103413                                  BCS     10$             ; IF CS, LESS THAN 3 CHARS CONVERTED
 000420 005001                                  CLR     R1              ; PERIODS NOT ALLOWED
 000422 004767  000000G                         CALL    $CAT5           ; CONVERT REST OF NODE NAME TO RAD50
 000426 010167  000002G                         MOV     R1,$NODNA+2     ; STORE SECOND HALF OF NODE NAME
 000432 022767  000006   000000G                CMP     #6,.PSTCN       ; IS THE NODE NAME TOO BIG ?
 000440 103002                                  BHIS    10$             ; IF HIS, NO - OKAY
 000442 000261                                  SEC                     ; ELSE, INDICATE FAILURE
 000444 000401                                  BR      20$             ; AND LEAVE
 000446 000241                          10$:    CLC                     ; INDICATE SUCCESS
 000450 000207                          20$:    RETURN
```

G 8

```
**FILE**ID**PRSCIR
```

```
PPPPPPPP     RRRRRRRR       SSSSSSSS     CCCCCCCC     IIIIII     RRRRRRRR
PPPPPPPP     RRRRRRRR       SSSSSSSS     CCCCCCCC     IIIIII     RRRRRRRR
PP      PP   RR      RR   SS           CC               II       RR      RR
PP      PP   RR      RR   SS           CC               II       RR      RR
PP      PP   RR      RR   SS           CC               II       RR      RR
PP      PP   RR      RR   SS           CC               II       RR      RR
PPPPPPPP     RRRRRRRR       SSSSSS     CC               II       RRRRRRRR
PPPPPPPP     RRRRRRRR       SSSSSS     CC               II       RRRRRRRR
PP           RR  RR               SS   CC               II       RR  RR
PP           RR  RR               SS   CC               II       RR  RR
PP           RR    RR             SS   CC               II       RR    RR         ....
PP           RR    RR             SS   CC               II       RR    RR         ....
PP           RR      RR   SSSSSSSS       CCCCCCCC     IIIIII     RR      RR       ....
PP           RR      RR   SSSSSSSS       CCCCCCCC     IIIIII     RR      RR       ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

H 8

PRSCIR - PARSE A CIRCUIT-ID MACRO V05.03b Saturday 29-Jun-85 13:20

```

000664  120305          20$:    CMPB    R3,R5           ; IS THE CHARACTER A LEGAL TERMINATOR ?
000666  001403                  BEQ     40$             ; IF EQ, YES - RETURN SUCCESS CODE
000670  000261                  SEC                     ; OTHERWISE INDICATE AN ERROR
000672  000402                  BR      50$             ;

000674  010616          30$:    MOV     SP,(SP)         ; MARK THE END OF THE STRING

000676  000241          40$:    CLC                     ; INDICATE SUCCESSFUL CONVERSION
000700  012602          50$:    MOV     (SP)+,R2        ; RESTORE REGISTERS
000702  012605                  MOV     (SP)+,R5        ; SET EXIT CONDITION CODES
000704                          RETURN                  ; RETURN TO CALLER
```

```
                                                                  G 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE            REFERENCES

LF.TIM  = 000010          #5-65
LF.UNL  = 020000          #5-65
LF.X2P  = 000000          #5-65
LN.CLO  = 000000          #5-65
LN.DUM  = 000005          #5-65
LN.LOA  = 000004          #5-65
LN.LOO  = 000003          #5-65
LN.OAU  = 000003          #5-65
LN.OFF  = 000001          #5-65
LN.ON   = 000000          #5-65
LN.OOP  = 000004          #5-65
LN.OPE  = 000001          #5-65
LN.REF  = 000002          #5-65
LN.SER  = 000002          #5-65
LN.STA  = 000017          #5-65
LN.SUB  = 000360          #5-55
LN.TRI  = 000006          #5-65
LP$MPT  = 000010          #6-74
LP$MUX  = 000004          #6-74      8-180      8-192      13-481
LP$TRB  = 000002          #6-74      8-228      8-236      14-571
LP$UNT  = 000001          #6-74      8-202      8-210
LP$WCN  = 000040          #6-74      6-74       8-185      8-206      8-232      14-544     14-569
LP$WDV  = 000020          #6-74      6-74
LP$WLD  = 000360          #6-74
LP$WTR  = 000200          #6-74      6-74       8-228      14-577
LP$WUN  = 000100          #6-74      6-74       8-202      8-232      14-548     14-569
L$ADJ     000110          #5-74
L$BUF     000060          #5-74
L$CHN     000054          #5-74
L$COU     000114          #5-74
L$CTB     000053          #5-74
L$CTL     000002          #5-74     *8-191      14-546
L$DDM     000000          #5-74     *8-173      15-628
L$FLG     000064          #5-74
L$FLX     000070          #5-74
L$LEN     000124          #5-74
L$LTM     000070          #5-74
L$MSG     000056          #5-74
L$MTYP    000067          #5-74
L$NAM     000000          #5-74      9-272      9-278
L$NLEN  = 000020          #5-74      5-74       5-74       9-273
L$NOD     000112          #5-74
L$NXT     000044          #5-74
L$OPT     000062          #5-74
L$PAR     000120          #5-74
L$PDV     000055          #5-74      8-133      8-157
L$PFG     000041          #5-74      8-180     *8-185      8-192     *8-202      8-206     *8-210     *8-228      8-232
                         *8-236    *13-481      14-544     14-548     14-569     14-571     14-577

L$PLB     000116          #5-74
L$PRO     000067          #5-74
L$PVC     000004          #5-74
L$SCN     000020          #5-74
```

```
                                                                  H 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 3
```

```
                                                  .SBTTL  .EXNC - SET NUMBER OF NODE COUNTERS
                                          ;+
                                          ; *** - .EXNN - SET NUMBER OF NODE COUNTERS
                                          ;
                                          ; INPUTS:
                                          ;         .PNUMB = THE NUMBER OF NODE COUNTERS
                                          ;
                                          ; OUTPUTS:
                                          ;         $NCOU = THE NUMBER OF NODE COUNTERS
                                          ;         THE  TRANSITION IS ACCEPTED/REJECTED.
                                          ;
                                          ;-

000360                                    .EXNC::
000360  016767  000000G 000000G                   MOV     .PNUMB,$NCOU      ; STORE THE MAX. NODE NUMBER
000366  026727  000000G 000002                    CMP     .PNUMB,#2         ; IS THE VALUE IN RANGE ?
000374  103426                                    BLO     40$               ; IF LO, OUT OF RANGE
000376  016700  000000G                           MOV     $NN,R0            ; GET MAX NODE ADDRESS FROM PARSE
000402  032767  000000G 000000G                   BIT     #OE.NN,$OPRND     ; MAX ADDRESS SPECIFIED IN THIS COMMAND LINE ?
000410  001015                                    BNE     30$               ; IF NE, SPECIFIED IN PARSE - USE IT
000412  012700  000000G                           MOV     #$TMLST,R0        ; ELSE POINT TO TEMPLATE LIST
000416  011000                            10$:    MOV     (R0),R0           ; POINT TO NEXT TEMPLATE
000420  001407                                    BEQ     20$               ; IF EQ, NO ROUTING INFO - MUST BE END NODE
000422  122760  000000G 000000G                   CMPB    #CS.ROU,C.STS(R0) ; IS THIS THE ROUTING TEMPLATE ?
000430  001372                                    BNE     10$               ; IF NE, NO
000432  016000  000004                            MOV     C.ADD(R0),R0      ; ELSE GET MAX NODE ADDRESS
000436  001002                                    BNE     30$               ; IF NE, GOT ONE
000440  012700  000000G                   20$:    MOV     #MAXNN,R0         ; ELSE IT'S AN END NODE
000444  026700  000000G                   30$:    CMP     .PNUMB,R0         ; CHECK HIGH END OF RANGE
000450  101405                                    BLOS    50$               ; IF LOS, OK
000452  062716  000002                    40$:    ADD     #2,(SP)           ; ELSE, REJECT TRANSITION
000456  012767  000000G 000000G                   MOV     #ERRILN,$ERROR    ; INVALID NUMBER
000464  000207                            50$:    RETURN
```

```
                                                                 G 12
PRSEXE     CREATED BY  MACRO  ON 29-JUN-85 AT 05:18     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE           REFERENCES

.EXNC      000360 RG     #10-238
.EXNN      000274 RG     #9-203
.EXOUT     001176 RG     #19-483
.EXRET     001236 RG     #20-504
.EXSB2     000552 RG     #12-309
.EXSEG     001044 RG     #16-422
.EXSUB     000544 RG     #12-305
.PFLAG   = ******   GX   *5-108
.PNUMB   = ******   GX    6-138       6-140       7-160       7-162       8-182       8-184       9-211       10-239      10-240
                          10-253      12-314      12-316      13-340      13-342      13-347      13-349      13-354      13-356
                          14-383      15-402      15-404      16-423      17-442      17-446      18-463      18-467      19-484
                          19-488      20-505      20-509
.PNUMH   = ******   GX    12-312      13-359      14-385      15-406      16-425      17-444      18-465      19-486      20-507
.PSTCN   = ******   GX   *5-87        5-89       *5-96        5-115
.PSTPT   = ******   GX   *5-86        5-88        5-114
```

```
                                                                 H 12
PRSEXE     CREATED BY  MACRO  ON 29-JUN-85 AT 05:18     PAGE 3
```

```
                                                     .SBTTL  CHKMUX - CHECK IF A DEVICE IS A MULTIPLEXOR
                                          ;+
                                          ; **-CHKMUX-CHECK IF A DEVICE IS A MULTIPLEXOR
                                          ;
                                          ; THIS ROUTINE IS CALLED TO DETERMINE HOW TO PARSE A LINE-ID.
                                          ;
                                          ; INPUTS:
                                          ;       R5 - ADDRESS OF CONTEXT AREA WITN
                                          ;       L$DDM(R5) = DDM NAME IN RAD50
                                          ;
                                          ; OUTPUTS:
                                          ;       IF LP$MUX SET IN L$PFG(R5), DEVICE IS MUX
                                          ;-

 000476                                   CHKMUX:
 000476                                           SWSTK$  10$                     ;; ENTER SYSTEM STATE
 000502                                           CALL    PDVAD                   ;; FIND IT'S PDV ADDRESS
 000506  103407                                   BCS     10$                     ;; IF CS, ILLEGAL DDM
 000510  132764  000040  000010                   BITB    #ZF.MUX,Z.FLG(R4)       ;; IS IT A MUX DEVICE ?
 000516  001403                                   BEQ     10$                     ;; IF EQ, NO - RETURN
 000520  152765  000004  000041                   BISB    #LP$MUX,L$PFG(R5)       ;; REMEMBER MUX DEVICE
 000526                                   10$:    RETURN
```

```
PRSLIN     CREATED BY  MACRO  ON 29-JUN-85 AT 13:23     PAGE 5
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME      REFERENCES

CALL              7-91      7-112     7-120     7-124     7-130     7-141     7-149     7-158     8-215     9-276
                 10-310    10-311    11-357    11-361    12-410
CIRCX$          #5-57      5-63
MANDF$          #5-57      5-61
PDVDF$          #5-57      5-59
RETC            #5-57     11-387
RETURN            7-166     8-236     9-293    10-316    11-388    12-420    13-465
SLTDF$          #5-57      5-60
SOB               7-95
SWSTK$           10-310    11-357
```

FILEID**PRSLOG

```
PRSLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME      REFERENCES
PRINT           #4-60
```

```
**FILE**ID**PRSMOD
```

```
                                                .SBTTL  .MOMXR - PARSE MAXIMUM RESTARTS
                                        ;+
                                        ; ** .MOMXR - PARSE MAXIMUM RESTARTS
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $MXR = PARSED MAXIMUM RESTARTS
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-
000350                                  .MOMXR::
000350  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
000354  001054                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
000356  105767  000001G                         TSTB    .PNUMB+1        ; TEST HIGH BYTE
000362  001051                                  BNE     ILLNUM          ; BR IF  ILLEGAL NUMBER
000364  116767  000000G 000000G                 MOVB    .PNUMB,$MXR     ; MOVE VALUE TO PARSED AREA
000372  000207                                  RETURN

```

```
                                                                      H  1
CFETSB      CREATED BY  MACRO  ON 29-JUN-85 AT 05:15      PAGE 1

SYMBOL  CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

BYTFLG     000006 R        #4-67       *5-92       *5-96       5-104
REJECT     000200 R          5-98        5-100       5-102     5-138      #5-165
$CLERR     000206 RG       #5-179
$GTNMB     000010 RG       #5-91
$GTNUM     000016 RG       #5-95
$GTR50     000072 RG       #5-133
.CAT5   =  ******  GX        5-155
.ERROR     000000 RG       #4-61       *5-180
.FLAGS     000002 RG       #4-62
.PNUMB  =  ******  GX        5-99        5-101       5-106     5-108
.PNUMH  =  ******  GX        5-97
.PSTCN  =  ******  GX        5-137       5-142       5-148
.PSTPT  =  ******  GX        5-140
.TEMP      000004 RG       #4-63         5-103       5-146
```

```
                                                                      I  1
**FILE**ID**CFEUPD
```

```
                                ;+
                                ; *** - $CNVD - CONVERT BINARY TO DECIMAL MAGNITUDE
                                ; *** - $CNVO - CONVERT BINARY TO OCTAL MAGNITUTDE
                                ;
                                ; INPUT:
                                ;       R0 = ADDRESS OF OUTPUT BUFFER
                                ;       R1 = NUMBER TO CONVERT
                                ;
                                ; OUTPUT:
                                ;       R0 = NEXT LOCATION IN OUTPUT BUFFER
                                ;       R1 = PRESERVED
                                ;       R2 = DESTROYED
                                ;
                                ;-

                                        .ENABL  LSB
000526  005002                  $CNVD:: CLR     R2              ; ZERO SUPPRESSION
000530  004767  000000G                 CALL    $CBDMG          ; CONVERT NUMBER TO DECIMAL ASCII
000534  112720  000056                  MOVB    #'.,(R0)+       ; APPEND A DECIMAL POINT
000540  000403                          BR      10$             ; AND LEAVE

000542  005002                  $CNVO:: CLR     R2              ; ZERO SUPPRESSION
000544  004767  000000G                 CALL    $CBOMG          ; CONVERT NUMBER TO OCTAL ASCII
000550  000207                  10$:    RETURN                  ; RETURN TO CALLER
                                        .DSABL  LSB
```

```
    151                                             .SBTTL  $PTOPT - CHECK FOR OPTIONAL PARAMETER
    152                                     ;+
    153                                     ; ** $PTOPT - CHECK FOR OPTIONAL PARAMETER
    154                                     ;
    155                                     ; INPUTS        R2 = OPTION BIT
    156                                     ;               .FLAG = ADDRESS OF TEMPLATE STATUS WORD
    157                                     ;
    158                                     ; OUTPUT        CARRY CLEAR: OPTION SELECTED
    159                                     ;               CARRY SET: OPTION NOT SELECTED, .COMMA IS UPDATED.
    160                                     ;-
    161
    162 000274                              $PTOPT::
    163 000274  030277  177502                      BIT     R2,@.FLAGS      ; CHECK FOR OPTION
    164 000300  000241                              CLC                     ; ASSUME SELECTED
    165 000302  001003                              BNE     10$             ; IF NE, SELECTED
    166 000304  005267  177476                      INC     .COMMA          ; ELSE REMEMBER UNSELECTED OPTION
    167 000310  000261                              SEC                     ; RETURN STATUS
    168 000312  000207                      10$:    RETURN
    169
    170
    171                                             .SBTTL  CHKOPT  - CHECK FOR PRECEDING OPTIONS
    172                                     ;+
    173                                     ; ** CHKOPT  - CHECK FOR PRECEDING OPTIONS
    174                                     ;
    175                                     ; INPUTS        .COMMA = COUNT OF PRECEDING UNPRINTED OPTIONS ON LINE
    176                                     ;
    177                                     ; OUTPUT        COMMAS ARE OUTPUT TO ACCOUNT FOR ANY PRECEDING
    178                                     ;               UNPRINTED OPTIONS.
    179                                     ;
    180                                     ;               .COMMA ZEROED
    181                                     ;-
    182
    183 000314  005267  177466              CHKOPT: INC     .COMMA          ; GET REAL COUNT OF COMMAS
    184 000320  005367  177462              10$:    DEC     .COMMA          ; REDUCE COUNT
    185 000324  100406                              BMI     20$             ; IF MI, NO MORE COMMAS
    186 000326  112777  000054  000000G             MOVB    #',,@$OBUF      ; BUFFER A COMMA
    187 000334  005267  000000G                     INC     $OBUF           ; BUMP BUFFER POINTER
    188 000340  000767                              BR      10$             ; CONTINUE
    189 000342  005267  177440              20$:    INC     .COMMA          ; RESET AND CLEAR CARRY
    190 000346  000207                              RETURN
    191
    192         000001                              .END
```

H 4

```
**FILE**ID**NOSNA




NN      NN    OOOOOO      SSSSSSSS  NN      NN    AAAAAA
NN      NN    OOOOOO      SSSSSSSS  NN      NN    AAAAAA
NN      NN  OO      OO  SS          NN      NN  AA      AA
NN      NN  OO      OO  SS          NN      NN  AA      AA
NNNN    NN  OO      OO  SS          NNNN    NN  AA      AA
NNNN    NN  OO      OO  SS          NNNN    NN  AA      AA
NN  NN  NN  OO      OO    SSSSSS    NN  NN  NN  AA      AA
NN  NN  NN  OO      OO    SSSSSS    NN  NN  NN  AA      AA
NN    NNNN  OO      OO          SS  NN    NNNN  AAAAAAAAAA
NN    NNNN  OO      OO          SS  NN    NNNN  AAAAAAAAAA
NN      NN  OO      OO          SS  NN      NN  AA      AA      ....
NN      NN  OO      OO          SS  NN      NN  AA      AA      ....
NN      NN    OOOOOO    SSSSSSSS    NN      NN  AA      AA      ....
NN      NN    OOOOOO    SSSSSSSS    NN      NN  AA      AA      ....




LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

I 4

NOSNA - STUB ACTION ROUTINES FO MACRO V05.03b Saturday 29-Jun-85 05:16 Page 3

```
Symbol table

P$ACC    000010 G         P$CPU    001664 G         P$LHE    003700 G         P$PCH    000644 G         P$SNV    006526 G
P$ACT    000072 G         P$CSR    001702 G         P$LIST   003460 G         P$PEN    001536 G         P$STA    007472 G
P$ADD    000124 G         P$CTM    001634 G         P$LMB    004000 G         P$PHA    005560 G         P$STI    006602 G
P$AID    000140 G         P$CUS    010130 G         P$LMX    004204 G         P$PHO    005606 G         P$STT    007504 G
P$AMC    000210 G         P$CVL    000600 G         P$LNID   003366 G         P$PID    005672 G         P$STY    007450 G
P$AMH    000236 G         P$DDT    001776 G         P$LOA    003500 G         P$PRI    005654 G         P$SUB    007534 G
P$AREA   003750 G         P$DDTH   001752 G         P$LOC    003554 G         P$PRO    003436 G         P$SWO    007556 G
P$ARID   000160 G         P$DEA    001722 G         P$LOO    003654 G         P$PST    005632 G         P$TER    007602 G
P$AST    000312 G         P$DES    002230 G         P$LOST   003632 G         P$RET    005740 G         P$TFIL   007720 G
P$ASTN   000352 G         P$DEV    002256 G         P$LOWN   003416 G         P$RFA    006072 G         P$TLO    007636 G
P$ASTO   000264 G         P$DFA    002112 G         P$MAD    003724 G         P$RMX    004510 G         P$TO     005466 G
P$BBT    000376 G         P$DIA    002276 G         P$MBN    004070 G         P$RRT    006042 G         P$TPAS   007676 G
P$BDF    002024 G         P$DLT    002134 G         P$MBR    004032 G         P$RST    006014 G         P$TRIB   010012 G
P$BFQ    005770 G         P$DST    002206 G         P$MBU    004132 G         P$RTI    006154 G         P$TRIG   010040 G
P$BLK    000426 G         P$DTE    002322 G         P$MCO    004260 G         P$RTPR   005714 G         P$TSTA   007740 G
P$BRT    000452 G         P$DUA    002362 G         P$MHO    004334 G         P$RTR    006120 G         P$TYP    000766 G
P$BSA    000024 G         P$DUC    002404 G         P$MLK    004400 G         P$SCO    006240 G         P$USE    010150 G
P$BSD    002476 G         P$DUM    002424 G         P$MLN    004156 G         P$SCT    006204 G         P$VEC    010200 G
P$BSI    003146 G         P$DUMO   002442 G         P$MOID   004750 G         P$SDU    006252 G         P$VEST   010214 G
P$BUS    000514 G         P$DUP    002462 G         P$MRB    004454 G         P$SDV    006502 G         P$WDF    002056 G
P$CAT    000552 G         P$DWE    002164 G         P$MRT    004562 G         P$SEAL   006634 G         P$WIT    010102 G
P$CHN    000620 G         P$DYTH   002542 G         P$MUL    005002 G         P$SEC    006336 G         P$XMT    007760 G
P$CIID   000666 G         P$ETY    005150 G         P$MVI    004672 G         P$SECI   006662 G         P$ZER    010244 G
P$CIST   000744 G         P$EVE    002566 G         P$MXB    004636 G         P$SELI   006734 G         X        000000RG      002
P$CLCI   001010 G         P$FNAM   002620 G         P$MXCO   004230 G         P$SELO   006762 G         Y        000000RG      003
P$CLE    001042 G         P$GNM    002656 G         P$MXLI   004356 G         P$SENO   007174 G         $ALPHA=  000022
P$CLK    001502 G         P$GRP    002636 G         P$MXR    004422 G         P$SEOB   007222 G         $ANY  =  000020
P$CLLI   001066 G         P$GTY    010064 G         P$MXRT   004612 G         P$SEPR   007252 G         $BLANK=  000006
P$CLLO   001116 G         P$HAD    002726 G         P$MXW    004716 G         P$SER    006560 G         $DIGIT=  000024
P$CLNO   001342 G         P$HAO    002754 G         P$NAM    005106 G         P$SERO   006426 G         $DNUMB=  000014
P$CLOB   001372 G         P$HBT    003032 G         P$NCP    000000 G         P$SET    006712 G         $EOS  =  000012
P$CLT    001424 G         P$HOS    003064 G         P$NET    005034 G         P$SETR   007302 G         $EXIT =  000000
P$CLTR   001452 G         P$HPAS   003000 G         P$NID    005056 G         P$SEXA   007012 G         $FAIL =  177777
P$CLXA   001150 G         P$HTM    002700 G         P$NOST   005124 G         P$SEXP   007046 G         $GPRM =  000001
P$CLXP   001206 G         P$IAT    003250 G         P$NUM    002342 G         P$SE5S   007104 G         $LAMDA=  000000
P$CL5S   001246 G         P$IDE    003122 G         P$OBJ    005234 G         P$SE9S   007140 G         $NUMBR=  000002
P$CL9S   001304 G         P$INA    000044 G         P$OBNA   005266 G         P$SGZ    006376 G         $RAD50=  000016
P$CMB    004302 G         P$IND    002516 G         P$OBUS   005330 G         P$SHO    007330 G         $RONLY=  000001
P$CMK    000534 G         P$INI    003170 G         P$OBVE   005350 G         P$SID    007430 G         $STRNG=  000004
P$COM    001522 G         P$INTH   003220 G         P$OID    005306 G         P$SIN    007372 G         $SUBXP=  000010
P$CON    001562 G         P$ITI    003304 G         P$OQL    005400 G         P$SLI    006452 G         $$PRMP   000000RG      005
P$COP    001606 G         P$LDOP   003532 G         P$OTI    005434 G         P$SLO    006304 G         $$$FLG=  177777
P$COS    001622 G         P$LEN    003336 G         P$PAR    005516 G         P$SMX    004534 G         $$$KEY=  177777
P$COU    005200 G         P$LGID   003604 G         P$PAS    005542 G         P$SNAM   007354 G         $$$LEN=  000022
P$COWN   000722 G

. ABS.   010270     000   (RW,I,GBL,ABS,OVR)
         000000     001   (RW,I,LCL,REL,CON)
$STATE   000000     002   (RO,D,LCL,REL,CON)
$KTAB    000000     003   (RO,D,LCL,REL,CON)
$KSTR    000000     004   (RO,D,LCL,REL,CON)
$PRMPT   010270     005   (RO,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics
```

```
PRMPTO - PROMPTS, OFFSET ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:02  Page 6-5
Symbol table

Work  file  reads: 0
Work  file writes: 0
Size of work file: 8759 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:23.02
SY:PRMPTO.V2,[135,134]PRMPTO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[135,10]V2,PROMPT
```

```
PRMPTO     CREATED BY  MACRO  ON 29-JUN-85 AT 17:02     PAGE 1
```

```
                                        .SBTTL  .RNINI - INITIALIZE NODE ADDRESS
                                ;+
                                ; *** .RNINI - INITIALIZE NODE ADDRESS
                                ;
                                ; INPUT:
                                ;       NONE.
                                ;
                                ; OUTPUT:
                                ;       $NODAD = 0
                                ;-
000452                          .RNINI::
000452  005067  000000G                 CLR     $NODAD                  ; EASY
000456  000207                          RETURN

                                        .SBTTL  .RNAD2 - SECOND PART OF NODE ADDRESS
                                ;+
                                ; *** - .RNAD2 - SECOND PART OF NODE ADDRESS
                                ;
                                ; INPUT:
                                ;       .PNUMB  = PARSED NUMBER
                                ;       $NODAD  = AREA AS SET UP BY .RNADD
                                ;
                                ; OUTPUT:
                                ;       PHASE 4 NODE ADDRESS.
                                ;-
000460                          .RNAD2::
000460  016702  000000G                 MOV     $NTADD,R2       ; PICK UP EXEC ADDRESS
000464  042702  001777                  BIC     #1777,R2        ; ISOLATE AREA
000470  005767  000000G                 TST     .PNUMH          ; 32 BIT VALUE ?
000474  001040                          BNE     10$             ; IF NE, ERROR
000476  016700  000000G                 MOV     .PNUMB,R0       ; ELSE PICK UP PARSED VALUE
000502  020027  001777                  CMP     R0,#1023.       ; CHECK HI END OF RANGE
000506  101033                          BHI     10$             ; IF HI, ERROR
000510  016701  000000G                 MOV     $NODAD,R1       ; PICK UP FIRST PART OF ADDRESS
000514  042701  176000                  BIC     #^C1777,R1      ; CLEAR "AREA"
000520  005701                          TST     R1              ; WAS THIS AREA ZERO ?
000522  001002                          BNE     7$              ; IF NE, NO
000524  010201                          MOV     R2,R1           ; ELSE WE WANT EXEC AREA
000526  000406                          BR      8$              ; KNOW IT'S VALID, SO GO FOLD IT IN
000530  020127  000077          7$:     CMP     R1,#63.         ; CHECK REAL AREA VALUE
000534  101020                          BHI     10$             ; IF HI, OUT OF RANGE
000536  000301                          SWAB    R1              ; SHIFT AREA NUMBER TO HI BYTE
000540  006301                          ASL     R1              ; MESS WITH IT ...
000542  006301                          ASL     R1              ;  .. A BIT (2 BITS, ACTUALLY)
000544  050100                  8$:     BIS     R1,R0           ; FOLD AREA INTO NODE ADDRESS
000546  010067  000000G                 MOV     R0,$NODAD       ; STORE IT
000552  020002                          CMP     R0,R2           ; IS IT LOCAL AREA + NODE 0 ?
000554  001404                          BEQ     15$             ; IF EQ, NON-STANDARD BUT WE'LL ALLOW IT
000556  032700  001777                  BIT     #1777,R0        ; ELSE IS IT OTHER NODE 0 ?
000562  001012                          BNE     20$             ; NO - SUCCESS !
000564  000404                          BR      10$             ; YES - NOT A VALID ADDRESS
000566  016767  000000G 000000G 15$:    MOV     $NTADD,$NODAD   ; DO WHAT HE MEANT, NOT WHAT HE SAID
000574  000405                          BR      20$
000576  062716  000002          10$:    ADD     #2,(SP)         ; REJECT THE TRANSITION
000602  012767  000000G 000000G         MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000610  000207                  20$:    RETURN
```

Table of contents

```
     5-    57      MACRO LIBRARY CALLS AND LOCAL DATA
     6-    74      BIT DEFINITIONS
     7-    77      LOCAL DATA
     8-    89      PRSCIR - PARSE AN ASCII CIRCUIT-ID
     9-   255      STOCIR - Store circuit-id in context area
    10-   286      NEWPVC - Check circuit syntax for new X.25 PVC
    11-   334      CAT5 - CONVERT ASCII TO RAD50
    12-   404      CDTB - CONVERT DECIMAL TO BINARY
    13-   461      CHKMUX - CHECK IF A DEVICE IS A MULTIPLEXOR
    14-   484      CHKDEV - CHECK VALIDITY OF DEVICE SPECIFIER
    15-   609      PDVAD - GET THE PDV ADDRESS
    16-   639      CHKWLD - CHECK FOR WILD CARD FIELD
```

```
                                          .SBTTL  CDTB - CONVERT DECIMAL TO BINARY
                                  ;+
                                  ; **-CDTB-CONVERT DECIMAL TO BINARY
                                  ;
                                  ; INPUTS:
                                  ;
                                  ;       R0 = ADDRESS OF THE FIRST CHARACTER TO CONVERT
                                  ;       R3 = LEGAL TERMINATING CHARACTER
                                  ;       R4 = ADDRESS OF THE END OF THE STRING + 1
                                  ;
                                  ; OUTPUTS:
                                  ;
                                  ;       C-BIT CLEAR:
                                  ;       R0 = ADDRESS OF THE NEXT CHARACTER IN THE STRING
                                  ;       R1 = CONVERTED RESULT
                                  ;       R2 = COMPLETION CODE (0 = MORE IN THE STRING)
                                  ;                            (NOT 0 = END OF THE STRING)
                                  ;
                                  ;       C-BIT SET:
                                  ;       ILLEGAL CHARACTER IN THE STRING
                                  ;
                                  ; REGISTERS MODIFIED:
                                  ;
                                  ;       NONE
                                  ;-

000706  005046                    CDTB:   CLR     -(SP)           ; SET COMPLETION FLAGS
000710  005001                            CLR     R1              ; CLEAR ACCUMULATION
000712  112002                    10$:    MOVB    (R0)+,R2        ; GET THE NEXT CHARACTER
000714  020400                            CMP     R4,R0           ; REACHED THE END OF THE STRING ?
000716  103421                            BLO     20$             ; IF LO, YES - EXIT WITH RESULT
000720  120302                            CMPB    R3,R2           ; IS THE CHARACTER A LEGAL TERMINATOR ?
000722  001415                            BEQ     15$             ; IF EQ, YES - EXIT WITH RESULT          ;04
000724  162702  000060                    SUB     #'0,R2          ; NORMALIZE THE CHARACTER
000730  020227  000012                    CMP     R2,#10.         ; IS IT WITHIN RANGE ?
000734  103016                            BHIS    40$             ; IF HIS, NO - EXIT WITH ERROR
000736  010046                            MOV     R0,-(SP)        ; SAVE THE STRING POINTER
000740  012700  000012                    MOV     #10.,R0         ; SET THE MULTIPLICAND
000744                                    CALL    $MUL            ; MULTIPLY 'EM UP
000750  012600                            MOV     (SP)+,R0        ; RESTORE THE STRING POINTER
000752  060201                            ADD     R2,R1           ; ACCUMULATE THE RESULT
000754  000756                            BR      10$             ; CONTINUE

000756  020400                    15$:    CMP     R4,R0           ; WAS 'LEGAL TERMINATOR' THE END-OF-STRING ?;04
000760  001001                            BNE     30$             ; IF NEQ, NO- EXIT WITH RESULT           ;04

000762  010616                    20$:    MOV     SP,(SP)         ; SET THE STRING END INDICATOR

000764  022701  000400            30$:    CMP     #256.,R1        ; IS THE RESULT WITH A BYTE QUANTITY ?
000770  101002                            BHI     50$             ; IF HI, YES - EXIT WITH RESULT
000772  000261                    40$:    SEC                     ; INDICATE ILLEGAL CHARACTER IN STRING
000774  000401                            BR      60$             ;   OR RESULT OUT OF RANGE

000776  000241                    50$:    CLC                     ; INDICATE SUCCESSFUL CONVERSION
001000  012602                    60$:    MOV     (SP)+,R2        ; SET COMPLETION CONDITIONS
001002                                    RETURN                  ; RETURN TO CALLER
```

```
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 3
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE              REFERENCES

L$SCR      000072           #5-74
L$SLEN     000066           #5-74
L$SLT      000042           #5-74      *10-325
L$SNM      000046           #5-74
L$SYL      000122           #5-74      *14-598     *14-599
L$TNM      000052           #5-74
L$TPT      000050           #5-74      *10-324
L$TRB      000040           #5-74      *8-237       14-579
L$TSZ      000070           #5-74
L$TYP      000063           #5-74
L$UNT      000003           #5-74      *8-211       14-550
L.COST     000015           #5-65
L.CTL      000012           #5-65       14-546
L.CVA      177776           #5-65
L.DDM      000002           #5-65       14-542
L.DDS      000004           #5-65
L.DLC      000003           #5-65       14-558
L.DLM      000006           #5-65
L.DLS      000010           #5-65
L.FLG      000000           #5-65       14-573      14-593
L.KRBA     000016           #5-65
L.LEN    = 000022           #5-65
L.MPF      000022           #5-65
L.NMST     000020           #5-65
L.NSTA     000014           #5-65       14-575      14-579      14-595
L.OWNR     000021           #5-65
L.UNT      000013           #5-65       14-550
ME.IID     177767             8-249     14-602
ME.UCO     177770             14-605
NEWPVC     000460 R           8-154    #10-308
N$$VCT   = ******             14-606
PDVAD      001334 R           13-477    14-527     #15-627
PDVID    = ******   GX        15-629
PDVTA    = ******   GX        14-559    15-632
PRSCIR     000004 RG        #8-123
PSIPT    = ******   GX        8-149     10-310
R$R0     = ******   GX       *8-245    *8-249      *14-602     *14-605
R$R1     = ******   GX       *8-166
R$$PRO   = ******             8-144
R$$RTR   = ******             8-137     14-507
R$$11D   = ******             5-64
R$$11M   = 000000             5-64
R$$11S   = ******             4-1       5-64        7-81        8-90        8-124       8-144       8-218       9-254       14-530
                              14-562    14-586
SF.ACT   = 000200           #5-65
SF.ENA   = 000100           #5-65
SF.LPB   = 000004           #5-65
SF.MFL   = 000040           #5-65
SF.PAC   = 000020           #5-65
SF.REA   = 000010           #5-65
SF.SER   = 000001           #5-65
SF.SVC   = 000002           #5-65
```

```
                                         .SBTTL  .EXHOS - STORE HOST NODE ADDRESS
                                 ;+
                                 ; *** - .EXHOS - STORE HOST NODE ADDRESS
                                 ;
                                 ; INPUT:
                                 ;       $NODAD =PARSED NODE ADDRESS
                                 ;
                                 ; OUTPUT:
                                 ;       $HOST = THE NODE ADDRESS
                                 ;       THE TRANSITION IS ALWAYS ACCEPTED
                                 ;
                                 ;-
000466                           .EXHO1::
000466  016767  000000G 000000G          MOV     $NODAD,$HOST    ; A LITTLE CHICANERY, HERE
000474  000207                           RETURN

000476                           .EXHOS::
000476  016700  000000G                  MOV     $NODAD,R0       ; KEEP YOUR EYES
000502  016767  000000G 000000G          MOV     $HOST,$NODAD    ;  ...ON THE NODE
000510  010067  000000G                  MOV     R0,$HOST        ;  ...
000514  001407                           BEQ     5$              ; IF EQ, EXECUTOR PARSED
000516  016700  000000G                  MOV     $NTADD,R0       ; ELSE PICK UP EXEC ADDRESS
000522  042700  001777                   BIC     #1777,R0        ; CLEAR NODE NUMBER
000526  020067  000000G                  CMP     R0,$HOST        ; IS THIS EXEC AREA, NODE ZERO ?
000532  001003                           BNE     10$             ; IF NE, NO

000534  016767  000000G 000000G  5$:     MOV     $NTADD,$HOST    ; ELSE DO WHAT HE MEANT, NOT WHAT HE SAID

000542  000207                   10$:    RETURN

```

```
PRSEXE     CREATED BY  MACRO  ON 29-JUN-85 AT 05:18     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

PRINT            #4-57
```

```
**FILE**ID**PRSLIN
```

```
                                                  .SBTTL  CHKDEV - CHECK VALIDITY OF DEVICE SPECIFIER
                                          ;+
                                          ; **-CHKDEV-CHECK VALIDITY OF DEVICE SPECIFIER
                                          ;
                                          ; THIS ROUTINE MAKES SURE THAT THE DEVICE SPECIFIED IS KNOWN TO THE SYSTEM,
                                          ; AND THAT THE FIELDS SPECIFIED APPLY.
                                          ;
                                          ; INPUTS:
                                          ;       R5 = ADDRESS OF CONTEXT AREA WITH
                                          ;       L$DDM(R5) = DDM NAME OF LINE IN RAD50
                                          ;       L$CTL(R5) = CONTROLLER NUMBER
                                          ;       L$UNT(R5) = UNIT NUMBER
                                          ;       L$TRB(R5) = TRIBUTARY NUMBER
                                          ;       L$PFG(R5) = PARSE FLAGS
                                          ;
                                          ; OUTPUTS:
                                          ;       IF CC, LINE IS FOUND AND IS LEGAL
                                          ;       ELSE, UNRECOGNIZED LINE OR ILLEGAL LINE-ID (ERROR CODE IN R0)
                                          ;
                                          ; NO REGISTERS MODIFIED
                                          ;-

000530                                    CHKDEV:
                                                  .IF DF  S$$BAS ! R$$RTR

                                                  CMP     #^RLC ,L$DDM(R5)          ; is it a PAM line card ?
                                                  BNE     5$                        ; if NE, no
                                                  SEC                               ; assume failure...
                                                  MOV     #ME.IID,R0                ; ...invalid id
                                                  BITB    #LP$TRB!LP$UNT,L$PFG(R5); trib or unit parsed ?
                                                  BNE     100$                      ; if NE, yes - error
                                                  MOV     #ME.UCO,R0                ; assume unrecognized component
                                                  BITB    #LP$WCN,L$PFG(R5)         ; wildcard unit parsed ?
                                                  BEQ     100$                      ; if EQ, not "LC-*"
                                                  MOV     #^RPAM,L$DDM(R5)          ; else look like "PAM-*-*"...
                                                  BISB    #LP$MUX!LP$UNT!LP$WCN!LP$WUN,L$PFG(R5) ; ...internally

                                                  .ENDC

000530                                    5$:     SWSTK$  100$                      ;: ENTER SYSTEM STATE
                                          ;
                                          ; MAKE SURE THE PROCESS EXISTS, AND IS A DDM
                                          ;
000534                                            CALL    PDVAD                     ;: FIND IT'S PDV ADDRESS
000540  103437                                    BCS     90$                       ;: IF CC, LINE NOT IN SYSTEM
                                          ;
                                          ; MAKE SURE THE LINE EXISTS
                                          ;
000542  017700  000000G                           MOV     @SLTNM,R0                 ;: GET NUMBER OF SYSTEM LINES    ; RJK02
000546  017703  000000G                           MOV     @SLTMA,R3                 ;: GET ADDRESS OF SYSTEM LINE IN; RJK02
000552  012301                            10$:    MOV     (R3)+,R1                  ;: GET ADDR OF SYSTEM LINE TABLE
000554  126102  000002                            CMPB    L.DDM(R1),R2              ;: DOES THIS PROCESS MATCH ?
000560  001020                                    BNE     20$                       ;: IF NE, NO - KEEP LOOKING
000562  132765  000040  000041                    BITB    #LP$WCN,L$PFG(R5)         ;: WILD CONTROLLER ?
000570  001033                                    BNE     100$                      ;: IF NE YES - SKIP LOOKING FOR SLT
000572  126165  000012  000002                    CMPB    L.CTL(R1),L$CTL(R5)       ;: DOES THIS CONTROLLER NUMBER MATCH ?
000600  001010                                    BNE     20$                       ;: IF NE, NO - KEEP LOOKING
```

```
**FILE**ID**PRSLOG




PPPPPPPP     RRRRRRRR       SSSSSSSS   LL             000000       GGGGGGGG
PPPPPPPP     RRRRRRRR       SSSSSSSS   LL             000000       GGGGGGGG
PP      PP   RR      RR   SS           LL           00    00   GG
PP      PP   RR      RR   SS           LL           00    00   GG
PP      PP   RR      RR   SS           LL           00    00   GG
PP      PP   RR      RR   SS           LL           00    00   GG
PPPPPPPP     RRRRRRRR       SSSSSS     LL           00    00   GG
PPPPPPPP     RRRRRRRR       SSSSSS     LL           00    00   GG
PP           RR  RR               SS   LL           00    00   GG  GGGGGG
PP           RR  RR               SS   LL           00    00   GG  GGGGGG
PP           RR    RR             SS   LL           00    00   GG      GG        ....
PP           RR    RR             SS   LL           00    00   GG      GG        ....
PP           RR      RR   SSSSSSSS     LLLLLLLLLL     000000       GGGGGG          ....
PP           RR      RR   SSSSSSSS     LLLLLLLLLL     000000       GGGGGG          ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
**FILE**ID**PRSMOD




PPPPPPPP     RRRRRRRR       SSSSSSSS   MM      MM     OOOOOO     DDDDDDDD
PPPPPPPP     RRRRRRRR       SSSSSSSS   MM      MM     OOOOOO     DDDDDDDD
PP      PP   RR      RR   SS           MMMM  MMMM   OO      OO   DD      DD
PP      PP   RR      RR   SS           MMMM  MMMM   OO      OO   DD      DD
PP      PP   RR      RR   SS           MM  MM  MM   OO      OO   DD      DD
PP      PP   RR      RR   SS           MM  MM  MM   OO      OO   DD      DD
PPPPPPPP     RRRRRRRR       SSSSSS     MM      MM   OO      OO   DD      DD
PPPPPPPP     RRRRRRRR       SSSSSS     MM      MM   OO      OO   DD      DD
PP           RR  RR               SS   MM      MM   OO      OO   DD      DD
PP           RR  RR               SS   MM      MM   OO      OO   DD      DD
PP           RR    RR             SS   MM      MM   OO      OO   DD      DD        ....
PP           RR    RR             SS   MM      MM   OO      OO   DD      DD        ....
PP           RR      RR   SSSSSSSS     MM      MM     OOOOOO     DDDDDDDD          ....
PP           RR      RR   SSSSSSSS     MM      MM     OOOOOO     DDDDDDDD          ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
    253                                                 .SBTTL  .MORST - PARSE RESTART TIMER
    254                                         ;+
    255                                         ; ** .MORST - PARSE RESTART TIMER
    256                                         ;
    257                                         ; INPUTS        .TPARS VARIABLES.
    258                                         ;
    259                                         ; OUTPUT        $RST = PARSED RESTART TIMER
    260                                         ;               THE TRANSITION IS ACCEPTED/REJECTED.
    261                                         ;-
    262 000374                                  .MORST::
    263 000374  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
    264 000400  001042                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
    265 000402  105767  000001G                         TSTB    .PNUMB+1        ; TEST HIGH BYTE
    266 000406  001037                                  BNE     ILLNUM          ; BR IF  ILLEGAL NUMBER
    267 000410  116767  000000G 000000G                 MOVB    .PNUMB,$RST     ; MOVE VALUE TO PARSED AREA
    268 000416  000207                                  RETURN
    269
```

I 1

```
**FILE**ID**CFEUPD
```

```
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   UU      UU   PPPPPPPP     DDDDDDDD
  CCCCCCCC   FFFFFFFFFF   EEEEEEEEEE   UU      UU   PPPPPPPP     DDDDDDDD
CC           FF           EE           UU      UU   PP      PP   DD      DD
CC           FF           EE           UU      UU   PP      PP   DD      DD
CC           FF           EE           UU      UU   PP      PP   DD      DD
CC           FF           EE           UU      UU   PP      PP   DD      DD
CC           FFFFFFFF     EEEEEEEE     UU      UU   PPPPPPPP     DD      DD
CC           FFFFFFFF     EEEEEEEE     UU      UU   PPPPPPPP     DD      DD
CC           FF           EE           UU      UU   PP           DD      DD
CC           FF           EE           UU      UU   PP           DD      DD
CC           FF           EE           UU      UU   PP           DD      DD     ....
CC           FF           EE           UU      UU   PP           DD      DD     ....
  CCCCCCCC   FF           EEEEEEEEEE   UUUUUUUUUU   PP           DDDDDDDD       ....
  CCCCCCCC   FF           EEEEEEEEEE   UUUUUUUUUU   PP           DDDDDDDD       ....


LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

J 1

CFEUPD - UPDATE THE CETAB FILE MACRO V05.03b Saturday 29-Jun-85 05:15
Table of contents

```
                                         ;+
                                         ; *** - $CLNUP - CALCULATE THE LENGTH OF THE OUTPUT BUFFER
                                         ;
                                         ; INPUT:
                                         ;       $OBUF = POINTER TO END OF OUTPUT BUFFER
                                         ;       $BUFR = ADDRESS OF START OF OUTPUT BUFFER
                                         ;
                                         ; OUTPUT:
                                         ;       $LEN = LENGTH OF THE OUTPUT BUFFER
                                         ;
                                         ;-

000552                                   $CLNUP::
000552  016767  177226  177222                  MOV     $OBUF,$LEN      ; CALCULATE THE LENGTH OF THE OUTPUT BUFFER
000560  162767  000000G 177214                  SUB     #$BUFR,$LEN     ;       ...
000566  000207                                  RETURN
```

```
                                                                          I 3
CFEUSB - TEMPLATE UPDATE SUBROU MACRO V05.03b Saturday 29-Jun-85 05:16  Page 6-1
Symbol table

BYTFLG  000130R         $CNVO = ****** GX       $PTNMB  000132RG        $PTR50  000224RG        .ERROR  000000RG
CHKOPT  000314R         $C5TA = ****** GX       $PTNUM  000140RG        .BUFF   000010RG        .FLAGS  000002RG
$CNVD = ****** GX       $OBUF = ****** GX       $PTOPT  000274RG        .COMMA  000006RG        .TEMP   000004RG


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000350    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 77 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.45
SY:CFEUSB.V2,[132,134]CFEUSB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEUSB
```

```
                                                                          J 3
CFEUSB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 1
```

```
        .TITLE  NOSNA - STUB ACTION ROUTINES FOR NON-SNA SYSTEMS
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - STUB ACTION ROUTINES
;       USED TO SATISFY REFERENCES IN SYSTEMS WITHOUT
;       SUPPORT FOR SNA GATEWAY COMMANDS
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-83
;       CREATED
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                                                                                       I  5
PRMPTS - PROMPTS, STRING ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:01  Page 5-5
Symbol table

Work  file  reads: 0
Work  file writes: 0
Size of work file: 8914 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:29.38
SY:PRMPTS.V2,[135,134]PRMPTS/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[135,10]PROMPT
```

```
                                                                                       J  5
PRMPTS     CREATED BY  MACRO  ON 29-JUN-85 AT 17:01     PAGE 1
```

```
                                                                   I  6
PRMPTO      CREATED BY  MACRO  ON 29-JUN-85 AT 17:02      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

X           000000 RG       #6-90
Y           000000 RG       #6-90
$ALPHA  = 000022            #6-90
$ANY    = 000020            #6-90
$BLANK  = 000006            #6-90
$DIGIT  = 000024            #6-90
$DNUMB  = 000014            #6-90
$EOS    = 000012            #6-90
$EXIT   = 000000            #6-90
$FAIL   = 177777            #6-90
$GOFF   = 000001            #3-2         4-1          6-90
$GPRM   = 000001            #5-67        6-90
$LAMDA  = 000000            #6-90
$NUMBR  = 000002            #6-90
$RAD50  = 000016            #6-90
$RONLY  = 000001            #5-71        6-90         6-90         6-90         6-90
$STRNG  = 000004            #6-90
$SUBXP  = 000010            #6-90
$$$FLG  = 177777            #6-90
$$$KEY  = 177777            #6-90
```

```
                                                                   J  6
PRMPTO      CREATED BY  MACRO  ON 29-JUN-85 AT 17:02      PAGE 2
```

```

                                          .SBTTL  .RNADD - STORE NODE ADDRESS
                                          .SBTTL  .RNAD3 - CHECK PARSED NODE ADDRESS
                                  ;+
                                  ; *** - .RNADD - STORE NODE ADDRESS
                                  ; *** - .RNAD3 - CHECK PARSED NODE ADDRESS
                                  ;
                                  ; INPUT:
                                  ;       .PNUMB = NODE ADDRESS
                                  ;       .PNUMH = NODE ADDRESS (HIGH ORDER)
                                  ;
                                  ; OUTPUT:
                                  ;       $NODAD = THE NODE ADDRESS
                                  ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                  ;
                                  ;-
                                          .ENABL  LSB
000612                            .RNADD::
000612  005767  000000G                   TST     .PNUMH          ; IS IT A WORD VALUE ?
000616  001054                            BNE     10$             ; IF NE, NO .. ERROR
000620  016700  000000G                   MOV     .PNUMB,R0       ; PICK UP PARSED VALUE
000624  020027  001777                    CMP     R0,#1023.       ; IS NODE ADDRESS TOO HIGH ?
000630  016701  000000G                   MOV     $NTADD,R1       ; PICK UP EXEC ADDRESS
000634  042701  001777                    BIC     #1777,R1        ; ISOLATE AREA
000640  050100                            BIS     R1,R0           ; FOLD EXEC AREA INTO PARSED VALUE
000642  010067  000000G                   MOV     R0,$NODAD       ; STORE THE NODE ADDRESS
000646  000445                            BR      30$             ;
                                  ;
                                  ; PERFORM CHECKS ON PARSED NODE ADDRESS.  THIS IS FOR REMOTE NODE
                                  ; ADDRESSES ONLY.
                                  ;
000650                            .RNAD3::
000650  016701  000000G                   MOV     $NODAD,R1       ; CHECK PARSED NODE ADDRESS
000654  001406                            BEQ     50$             ; IF EQ, ITS THE EXEC ADDRESS
000656  016700  000000G                   MOV     $NTADD,R0       ; ELSE PICK UP EXEC ADDRESS
000662  042700  001777                    BIC     #1777,R0        ; ISOLATE AREA
000666  020001                            CMP     R0,R1           ; IS PARSED ADDRESS LOCAL AREA, NODE ZERO ?
000670  001003                            BNE     1$              ; IF NE, NO
000672  016767  000000G 000000G 50$:      MOV     $NTADD,$NODAD   ; ELSE DEFAULT TO EXEC
000700  026767  000000G 000000G 1$:       CMP     $NODAD,$NTADD   ; IS THIS THE EXECUTOR NODE ADDRESS ?
000706  001025                            BNE     30$             ; IF NE, NO .. OKAY
000710  022767  000000G 000000G 2$:       CMP     #CM.LIS,$CMAND  ; IS THIS THE LIST COMMAND ?
000716  001404                            BEQ     3$              ; IF EQ, YES .. OKAY
000720  022767  000000G 000000G           CMP     #CM.DEF,$CMAND  ; IS THIS THE DEFINE COMMAND?
000726  001004                            BNE     5$              ; IF NE, NO .. ERROR
000730  012767  000000G 000000G 3$:       MOV     #CO.EXE,$COMP   ; ELSE, SET TO EXECUTOR NODE
000736  000411                            BR      30$             ; ...
000740  012767  000105' 000000G 5$:       MOV     #ERREXE,$ERROR  ; EXECUTOR NODE NOT ALLOWED
000746  000403                            BR      20$             ; AND REJECT
000750  012767  000000G 000000G 10$:      MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000756  062716  000002          20$:      ADD     #2,(SP)         ; REJECT THE TRANSITION
000762  000207                  30$:      RETURN
                                          .DSABL  LSB
```

```
.IF NDF R$$11S
          .TITLE  PRSCIR - PARSE A CIRCUIT-ID
.IFF
          .TITLE  PRSCIR - PARSE LINE/CIRCUIT (RSX11S)
.ENDC
          .IDENT  /V05.00/
          .ENABL  LC
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983 1984, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; NETWORK MANAGEMENT - PARSE A CIRCUIT-ID STRING
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  08-JUL-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 2.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                                        .SBTTL  CHKMUX - CHECK IF A DEVICE IS A MULTIPLEXOR
                                ;+
                                ; **-CHKMUX-CHECK IF A DEVICE IS A MULTIPLEXOR
                                ;
                                ; THIS ROUTINE IS CALLED TO DETERMINE HOW TO PARSE A LINE-ID.
                                ;
                                ; INPUTS:
                                ;       R5 - ADDRESS OF CONTEXT AREA WITN
                                ;       L$DDM(R5) = DDM NAME IN RAD50
                                ;
                                ; OUTPUTS:
                                ;       IF LP$MUX SET IN L$PFG(R5), DEVICE IS MUX
                                ;
                                ;-

001004                          CHKMUX:
001004                                  SWSTK$  10$                     ;; ENTER SYSTEM STATE
001010                                  CALL    PDVAD                   ;; FIND IT'S PDV ADDRESS
001014  103407                          BCS     10$                     ;; IF CS, ILLEGAL DDM
001016  132764  000040  000010          BITB    #ZF.MUX,Z.FLG(R4)       ;; IS IT A MUX DEVICE ?
001024  001403                          BEQ     10$                     ;; IF EQ, NO - RETURN
001026  152765  000004  000041          BISB    #LP$MUX,L$PFG(R5)       ;; REMEMBER MUX DEVICE
001034                          10$:    RETURN
```

```
SYMBOL CROSS REFERENCE                                       CREF    04.00

SYMBOL  VALUE               REFERENCES

SF.UNL  = 000040            #5-65
SLTMA   = ******   GX        14-540
SLTNM   = ******   GX        14-539      14-598
SPACE   = 000040            #5-72         9-274
STOCIR    000402 RG           8-151      #9-268
S$$BAS  = ******              8-137      14-507
S.COST    000001            #5-65
S.FLG     000000            #5-65
S.LEN     000004            #5-65
S.NMST    000002            #5-65
S.OWNR    000003            #5-65
XPREF     000000 R          #7-84        7-85        10-319
XPREFL  = 000004            #7-85       10-316       10-318
X$$MCB  = ******              5-64       5-64
ZF.COU  = 001000            #5-64
ZF.DDM  = 000001            #5-64       15-634
ZF.DIA  = 004000            #5-64
ZF.DLC  = 000002            #5-64
ZF.DVP  = 100000            #5-64
ZF.INI  = 040000            #5-64
ZF.KMX  = 000020            #5-64
ZF.LLC  = 000004            #5-64
ZF.LMC  = 000100            #5-64
ZF.MAN  = 020000            #5-64
ZF.MFL  = 000010            #5-64
ZF.MTM  = 000400            #5-64
ZF.MUX  = 000040            #5-64       13-479
ZF.PSE  = 002000            #5-64
ZF.SLI  = 010000            #5-64
ZF.TIM  = 000200            #5-64
ZF.X3P  = 000000            #5-64
ZS.ASN  = 100000            #5-64
ZS.BSY  = 140000            #5-64
Z.AVL     000014            #5-64
Z.DAT     000016            #5-64
Z.DSP     000000            #5-64        5-64
Z.FLG     000010            #5-64       13-479       15-634
Z.LEN   = 000016            #5-64
Z.LLN     000006            #5-64
Z.MAP     000020            #5-64
Z.NAM     000004            #5-64       14-532       14-564
Z.PCB     000012            #5-64
Z.SCH     000007            #5-64
$FNPVC  = ******   GX         8-152
$HEADR  = ******   GX        14-606
$MUL    = ******   GX        11-381      12-442
$SAVAL  = ******   GX         8-131
```

```
                                       .SBTTL  .EXSUB  - PARSE EXECUTOR SUBADDRESS
                                       .SBTTL  .EXSB2 - PARSE EXECUTOR SUBADDRESS RANGE
                               ;+
                               ; ** .EXSUB  - PARSE EXECUTOR SUBADDRESS
                               ; ** .EXSB2 - PARSE EXECUTOR SUBADDRESS RANGE
                               ;
                               ; INPUTS:       .TPARS CONVENTIONS.
                               ;
                               ; OUTPUTS:      $ESUB = PARSED SUBADDRESS
                               ;               $ESUB+2 = SECOND PART OF SUBADDRESS RANGE OR -1 IF NONE.
                               ;
                               ;               THE TRANSITION IS ACCEPTED OR REJECTED.
                               ;-
                                       .ENABL  LSB

000544                         .EXSUB::
000544 012700  000000G                 MOV     #$ESUB,R0         ; POINT TO PARSED DATA AREA
000550 000402                          BR      10$               ; JOIN COMMON CODE

000552                         .EXSB2::
000552 012700  000002G                 MOV     #$ESUB+2,R0       ; POINT TO PARSED DATA AREA
000556 012767  177777  000002G 10$:    MOV     #-1,$ESUB+2       ; NO HIGH END YET
000564 005767  000000G                 TST     .PNUMH            ; 32 BIT VALUE
000570 001007                          BNE     20$               ; IF NE, YES - ERROR
000572 026727  000000G 023417          CMP     .PNUMB,#9999.     ; DO RANGE CHECK
000600 101003                          BHI     20$               ; IF HI, OUT OF RANGE
000602 016710  000000G                 MOV     .PNUMB,(R0)       ; STUFF IT IN PARSED DATA AREA
000606 000405                          BR      30$               ; SUCCESS !

000610 012767  000000G 000000G 20$:    MOV     #ERRILN,$ERROR    ; ILLEGAL NUMBER
000616 062716  000002                  ADD     #2,(SP)           ; REJECT TRANSITION

000622 000207                  30$:    RETURN

                                       .DSABL  LSB
```

I 12

```
**FILE**ID**PRSLIN




PPPPPPPP     RRRRRRRR       SSSSSSSS   LL             IIIIII     NN      NN
PPPPPPPP     RRRRRRRR       SSSSSSSS   LL             IIIIII     NN      NN
PP      PP   RR      RR   SS           LL               II       NN      NN
PP      PP   RR      RR   SS           LL               II       NN      NN
PP      PP   RR      RR   SS           LL               II       NNNN    NN
PP      PP   RR      RR   SS           LL               II       NNNN    NN
PPPPPPPP     RRRRRRRR       SSSSSS     LL               II       NN  NN  NN
PPPPPPPP     RRRRRRRR       SSSSSS     LL               II       NN  NN  NN
PP           RR  RR               SS   LL               II       NN    NNNN
PP           RR  RR               SS   LL               II       NN    NNNN
PP           RR    RR             SS   LL               II       NN      NN       ....
PP           RR    RR             SS   LL               II       NN      NN       ....
PP           RR      RR   SSSSSSSS     LLLLLLLLLL     IIIIII     NN      NN       ....
PP           RR      RR   SSSSSSSS     LLLLLLLLLL     IIIIII     NN      NN       ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

J 12

PRSLIN - PARSE A LINE-ID   MACRO V05.03b Saturday 29-Jun-85 13:23

```
000602  132765  000100  000041           BITB    #LP$WUN,L$PFG(R5)          ;; WILD UNIT ?
000610  001023                           BNE     100$                       ;; IF NE, YES - GO ON TO MUX TEST
000612  126165  000013  000003           CMPB    L.UNT(R1),L$UNT(R5)        ;; DOES THIS UNIT NUMBER MATCH ?
000620  001417                           BEQ     100$                       ;; IF EQ, FOUND THE CORRECT SLN
000622  005300                   20$:    DEC     R0                         ;; LOOK THROUGH ALL SYSTEM LINES
000624  001352                           BNE     10$                        ;; ...
000626  000404                           BR      90$                        ;; IF HERE, LINE NOT FOUND

000630  012766  177767  000002   80$:    MOV     #ME.IID,2(SP)              ;; SET USER R0 TO INVALID LIN-ID
000636  000403                           BR      95$

000640  012766  177770  000002   90$:    MOV     #ME.UCO,2(SP)              ;; SET USER R0 TO UNRECOGNIZED LINE
000646                           95$:    RETC    R0                         ;; SET/CLEAR USER CARRY
000660                           100$:   RETURN                             ;; AND RETURN TO USER MODE
```

Table of contents

```
     4-   55     LOCAL MACROS
     5-   66     CKEVE  - COMMON ROUTINE TO CHECK AND PROCESS AN EVENT NUMBER
     6-  114     CKEVEB - CHECK EVENT RANGE BEGINNING
     7-  148     CKEVEC - CHECK EVENT CLASS
     8-  191     CKEVEN - CHECK EVENT NUMBER
     9-  220     CKEVER - CHECK EVENT RANGE
```

Table of contents

```
     4-   55     .MOCUG - CONVERT USER GROUP TO RAD50
     5-   83     .MOCAT - PARSE CALL TIMER
     6-  101     .MOCLT - PARSE CLEAR TIMER
     7-  119     .MODFD - PARSE DEFAULT DATA
     8-  140     .MOMXD - PARSE MAXIMUM DATA
     9-  162     .MODFW - PARSE DEFAULT WINDOW
    10-  181     .MOMCL - PARSE MAXIMUM CLEARS
    11-  199     .MOMRE - PARSE MAXIMUM RESETS
    12-  217     .MORET - PARSE RESET TIMER
    13-  235     .MOMXR - PARSE MAXIMUM RESTARTS
    14-  253     .MORST - PARSE RESTART TIMER
    15-  271     .MOMC1 - PARSE MAXIMUM CIRCUITS (MODULE CHARACTERISTICS)
    16-  291     .MOMCI - PARSE MAXIMUM CIRCUITS (DTE)
    17-  318     .MONUM - PARSE GROUP NUMBER
    18-  337     .MOINC  - INITIALIZE CHANNEL LIST
    19-  352     .MOCNN  - PARSE CHANNEL NUMBER
    20-  381     .MOCRN  - MARK CHANNEL RANGE
    21-  400     .MODST - PARSE DESTINATION ID
    22-  428     .MOCLM  - PARSE CALL MASK
    22-  429     .MOCLV  - PARSE CALL VALUE
    23-  472     .MOPRI  - PARSE LIST INSERTION PRIORITY
    24-  490     .MODS1  - PARSE DESTINATION SUBADDRESS RANGE
    24-  491     .MODS2  - PARSE DESTINATION SUBADDRESS RANGE
    25-  522     .MOOBN  - PARSE OBJECT NUMBER
    26-  539     .MOOBJ  - PARSE OBJECT STRING
    27-  562     .MONNA  - PARSE NETWORK NAME
```

```
                                                        .SBTTL  .MOMC1 - PARSE MAXIMUM CIRCUITS (MODULE CHARACTERISTICS)
                                        ;+
                                        ; ** .MOMC1 - PARSE MAXIMUM CIRCUITS (MODULE CHARACTERISTICS)
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $MCI = PARSED MAXIMUM CIRCUITS
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;-

 000420                                 .MOMC1::
 000420  005767  000000G                        TST     .PNUMH          ; CHECK FOR HIGH ORDER PART
 000424  001030                                 BNE     ILLNUM          ; IF NE, OUT OF RANGE
 000426  026727  000000G 000377                 CMP     .PNUMB,#255.    ; CHECK HIGH BOUND
 000434  101024                                 BHI     ILLNUM          ; IF HI, OUT OF RANGE
 000436  016767  000000G 000000G                MOV     .PNUMB,$MCI     ; SAVE NUMBER
 000444  001420                                 BEQ     ILLNUM          ; IF EQ, TOO LOW
 000446  000207                                 RETURN
```

Table of contents

```
     4-   54     MACRO DEFINITIONS
     6-  191     $CFUPD - UPDATE THE CETAB FILE
     7-  256     SKIPIT - DON'T WRITE OUT THE LINE (ZERO $LEN)
     8-  272     $MOVIT - MOVE INPUT LINE INTO OUTPUT FILE WITHOUT EDITS
    13-  374     CLENUP - END$DF MACRO FOUND
    374                                         .SBTTL  CLENUP - END$DF MACRO FOUND
    375                                 ;+
    376                                 ; *** - CLENUP - CLEAN UP AFTER END$DF MACRO FOUND
    377                                 ;
    378                                 ; INPUT:
    379                                 ;       R3-R5 = TPARS REGISTERS
    380                                 ;
    381                                 ; OUTPUT:
    382                                 ;       CONTROL DISPATCHED TO ALL ROUTINES WHICH CAN ALLOCATE/DE-ALLOCATE
    383                                 ;       TEMPLATES.
    384                                 ;
    385                                 ;-
    386
    387 000570  010446                  CLENUP: MOV     R4,-(SP)        ; SAVE POINTER TO RESTOF LINE
    388 000572  005767  000000G                 TST     $PSIPT          ; PSI IN THIS SYSTEM ?
    389 000576  001407                          BEQ     10$             ; IF EQ, NO
    390 000600  004767  000000G                 CALL    $UPPSN          ; UPDATE THE PSI TEMPLATES
    391 000604  005767  000000G                 TST     $SVUPD          ; HAVE THE SVCS BEEN UPDATED ?
    392 000610  001002                          BNE     10$             ; IF NE, YES - ALREADY DONE
    393 000612  004767  000000G                 CALL    $UPSVC          ; UPDATE THE SVC TEMPLATES
    394 000616  004767  000000G         10$:    CALL    $UPOBJ          ; UPDATE THE OBJECT TEMPLATES
    395 000622  004767  000000G                 CALL    $UPEVT          ; UPDATE THE EVENT TEMPLATE
    396 000626  004767  000000G                 CALL    $UPREM          ; UPDATE THE REMOTE TEMPLATES
    397 000632                                  GTSK$S  #ARGLST         ; GET TASK PARAMETERS
    398 000644  012700  000012'                 MOV     #ARGLST+G.TSTN+4,R0 ; POINT TO UIC AREA
    399 000650  016701  177170                  MOV     ARGLST+G.TSDU,R1 ; ASSUME WE CAN ONLY GET CURRENT UIC
    400 000654                                  DIR$    #GINDPB         ; TRY TO GET LOGIN UIC
    401 000662  103402                          BCS     11$             ; WELL, AT LEAST WE TRIED
    402 000664  016701  177276                  MOV     GINBUF+G$LGUIC,R1 ; ELSE USE THE REAL LOGIN UIC
    403 000670  000301                  11$:    SWAB    R1              ; MAKE HI BYTE LOW, LOW BYTE HIGH
    404 000672  110120                          MOVB    R1,(R0)+        ; PUT (LOGIN) GROUP AFTER TASK ID
    405 000674  105020                          CLRB    (R0)+           ; KEEP HI BYTE CLEAN
    406 000676  000301                          SWAB    R1              ; GET USER BYTE LOW
    407 000700  110120                          MOVB    R1,(R0)+        ; DITTO FOR USER
    408 000702  105020                          CLRB    (R0)+           ; ...
    409 000704                                  GTIM$S  R0              ; GET TIME AFTER UIC
    410 000714  012600                          MOV     (SP)+,R0        ; POINT PAST END$DF IN BUFFER
    411 000716  012701  000052'                 MOV     #FMTSTR,R1      ; POINT TO CONTROL STRING
    412 000722  012702  000006'                 MOV     #ARGLST+G.TSTN,R2 ; POINT TO TASK ID
    413 000726  004767  000000G                 CALL    $EDMSG          ; FORMAT IT
    414 000732  162700  000000G                 SUB     #$BUFI,R0       ; CALCULATE NEW LENGTH
    415 000736  010067  177040                  MOV     R0,$LEN         ; SAVE IT
    416 000742  000207                          RETURN
    417
    418         000001                          .END
```

```
                                                                    J  3
CFEUSB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE             REFERENCES

BYTFLG     000130 R       #4-74       *5-95       *5-99        5-106
CHKOPT     000314 R        5-100       5-134      #6-183
$CNVD    = ******   GX     5-101
$CNVO    = ******   GX     5-104
$C5TA    = ******   GX     5-142
$OBUF    = ******   GX     5-112      *5-114       5-140      *5-145       6-186       *6-187
$PTNMB     000132 RG      #5-94
$PTNUM     000140 RG      #5-98
$PTOPT     000274 RG      #6-162
$PTR50     000224 RG      #5-133
.BUFF      000010 RG      #4-70
.COMMA     000006 RG      #4-69       *6-166      *6-183      *6-184      *6-189
.ERROR     000000 RG      #4-66
.FLAGS     000002 RG      #4-67        6-163
.TEMP      000004 RG      #4-68        5-105       5-139
```

```
                                                                    K  3
CFEUSB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 2
```

```
                                        ;+
                                        ; REJECT TPARS TRANSITION / PROVIDE NULL ROUTINE
                                        ;
                                        ; INPUTS:       NONE
                                        ;
                                        ; OUTPUTS:      ALL REGISTERS PRESERVED
                                        ;
                                        ; THIS ROUTINE IS USED TO REJECT ANY TPARS TRANSITIONS THAT
                                        ; SHOULD NOT BE TAKEN IN SYSTEMS WITHOUT SNA GATEWAY SUPPORT
                                        ;
                                        ; THOSE ENTRY POINTS WHICH SIMPLY LABEL THE RETURN INSTRUCTION
                                        ; MERELY SATISFY REFERENCES TO CALLS WHICH WILL NEVER BE TAKEN
                                        ;
                                        ;-

000000                                  $SNLIS::                ; ''LIST MODULE SNA-PROTOCOL''
000000                                  $SNDEF::                ; ''DEFINE MODULE SNA-PROTOCOL''
000000                                  $SNPUR::                ; ''PURGE MODULE SNA-PROTOCOL''
000000                                  $TMSNA::                ; ''SNA$DF'' INPUT
000000                                  $TMSAC::                ; ''SAC$DF'' INPUT
000000                                  $TMSAD::                ; ''SAD$DF'' INPUT
000000                                  $TMSPU::                ; ''SPU$DF'' INPUT
000000                                  $UPSNA::                ; ''SNA$DF'' UPDATE
000000                                  $UPSAC::                ; ''SAC$DF'' UPDATE
000000                                  $UPSAD::                ; ''SAD$DF'' UPDATE
000000                                  $UPSPU::                ; ''SPU$DF'' UPDATE

000000  062716  000002                          ADD     #2,(SP) ; REJECT TPARS TRANSITION
                                                                ;
000004                                  $SESNA::                ; PROCESS ''DEFINE'' COMMAND
000004                                  $SHSNA::                ;         ''LIST''
000004                                  $PUSNA::                ;         ''PURGE''
000004  000207                                  RETURN          ; RETURN

        000001                                  .END
```

```
PRMPTS     CREATED BY  MACRO  ON 29-JUN-85 AT 17:01       PAGE 1
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

X          000000 RG        #5-90
Y          000000 RG        #5-90
$ALPHA   = 000022           #5-90
$ANY     = 000020           #5-90
$BLANK   = 000006           #5-90
$DIGIT   = 000024           #5-90
$DNUMB   = 000014           #5-90
$EOS     = 000012           #5-90
$EXIT    = 000000           #5-90
$FAIL    = 177777           #5-90
$GOFF    = ******            3-1        5-90
$GPRM    = 000001           #4-67       5-90
$LAMDA   = 000000           #5-90
$NUMBR   = 000002           #5-90
$RAD50   = 000016           #5-90
$RONLY   = 000001           #4-71       5-90        5-90        5-90        5-90
$STRNG   = 000004           #5-90
$SUBXP   = 000010           #5-90
$$PRMP     000000 RG        #5-90
$$$FLG   = 177777           #5-90
$$$KEY   = 177777           #5-90
```

J 6

```
PRMPTO     CREATED BY  MACRO  ON 29-JUN-85 AT 17:02     PAGE 2
MACRO CROSS REFERENCE                                   CREF    04.00
MACRO NAME       REFERENCES

DBGTP$           #6-90
ISTAT$           #5-63      6-90
MTRAN$           #6-90
PRMPT$           #5-63      6-94      #6-96     #6-97     #6-98     #6-99     #6-100    #6-101    #6-102    #6-103
                 #6-104    #6-105    #6-106    #6-107    #6-108    #6-109    #6-110    #6-111    #6-112    #6-113
                 #6-114    #6-115    #6-116    #6-117    #6-118    #6-119    #6-120    #6-121    #6-122    #6-123
                 #6-124    #6-125    #6-126    #6-127    #6-128    #6-129    #6-130    #6-131    #6-132    #6-133
                 #6-134    #6-135    #6-136    #6-137    #6-138    #6-139    #6-140    #6-141    #6-142    #6-143
                 #6-144    #6-145    #6-146    #6-147    #6-148    #6-149    #6-150    #6-151    #6-152    #6-153
                 #6-154    #6-155    #6-156    #6-157    #6-158    #6-159    #6-160    #6-161    #6-162    #6-163
                 #6-164    #6-165    #6-166    #6-167    #6-168    #6-169    #6-170    #6-171    #6-172    #6-173
                 #6-174    #6-175    #6-176    #6-177    #6-178    #6-179    #6-180    #6-181    #6-182    #6-183
                 #6-184    #6-185    #6-186    #6-187    #6-188    #6-189    #6-190    #6-191    #6-192    #6-193
                 #6-194    #6-195    #6-196    #6-197    #6-198    #6-199    #6-200    #6-201    #6-202    #6-203
                 #6-204    #6-205    #6-206    #6-207    #6-208    #6-209    #6-210    #6-211    #6-212    #6-213
                 #6-214    #6-215    #6-216    #6-217    #6-218    #6-219    #6-220    #6-221    #6-222    #6-223
                 #6-224    #6-225    #6-226    #6-227    #6-228    #6-229    #6-230    #6-231    #6-232    #6-233
                 #6-234    #6-235    #6-236    #6-237    #6-238    #6-239    #6-240    #6-241    #6-242    #6-243
                 #6-244    #6-245    #6-246    #6-247    #6-248    #6-249    #6-250    #6-251    #6-252    #6-253
                 #6-254    #6-255    #6-256    #6-257    #6-258    #6-259    #6-260    #6-261    #6-262    #6-263
                 #6-264    #6-265    #6-266    #6-267    #6-268    #6-269    #6-270    #6-271    #6-272    #6-273
                 #6-274    #6-275    #6-276    #6-277    #6-278    #6-279    #6-280    #6-281    #6-282    #6-283
                 #6-284    #6-285    #6-286    #6-287    #6-288    #6-289    #6-290    #6-291    #6-292    #6-293
                 #6-294
```

K 6

```
**FILE**ID**PRSACT
```

```
                                                .SBTTL  .EXADD - STORE EXECUTOR NODE ADDRESS
                                        ;+
                                        ; *** - .EXADD - STORE EXECUTOR NODE ADDRESS
                                        ;
                                        ; INPUT:
                                        ;       $NODAD =PARSED NODE ADDRESS FROM .RNADD
                                        ;
                                        ; OUTPUT:
                                        ;       $NODAD = THE NODE ADDRESS
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-
                                                .ENABL  LSB
000764                                  .EXADD::
000764  012700  000000G                         MOV     #$TMLST,R0      ; GET START OF TEMPLATES
000770  011000                          5$:     MOV     (R0),R0         ; GET NEXT TEMPLATE
000772  001421                                  BEQ     30$             ; IF EQ, END OF TEMPLATES
000774  122760  000000G 000000G                 CMPB    #CS.REM,C.STS(R0) ; ELSE, IS THIS A REMOTE TEMPLATE ?
001002  001372                                  BNE     5$              ; IF NE, NO .. KEEP LOOKING
001004  026760  000000G 000010                  CMP     $NODAD,C.ADD(R0) ; DO ADDRESSES MATCH ?
001012  001366                                  BNE     5$              ; IF NE, NO
001014  012767  000041' 000000G                 MOV     #ERRADD,$ERROR  ; ELSE, ADDRESS ALREADY IN USE
001022  000403                                  BR      20$             ; AND REJECT
001024                                  ILLNUM:
001024  012767  000000G 000000G         10$:    MOV     #ERRILN,$ERROR  ; INVALID NUMBER
001032  062716  000002                  20$:    ADD     #2,(SP)         ; REJECT THE TRANSITION
001036  000207                          30$:    RETURN

                                                .DSABL  LSB
```

```
     57                                        .SBTTL  MACRO LIBRARY CALLS AND LOCAL DATA
     58                                 ;
     59                                 ; MACRO LIBRARY CALLS
     60                                 ;
     61                                        .MCALL  PDVDF$,SLTDF$,MANDF$,RETC,PHBDF$,PVCDF$,CIRCX$
     62                                        .MCALL  SAVRG,RESRG
     63
     64 000000                                 PDVDF$                          ; DEFINE PDV OFFSETS
     65 000000                                 SLTDF$                          ; DEFINE SLT OFFSETS
     66 000000                                 MANDF$                          ; NICE PROTOCOL SYMBOLS
     67 000000                                 PHBDF$                          ; PSI home block offsets
     68 000000                                 PVCDF$                          ; DEFINE PVC NAME BLOCK OFFSETS
     69                                 ;
     70                                 ; LOCAL SYMBOL DEFINITIONS
     71                                 ;
     72             000040                     SPACE = 40                      ; ASCII SPACE
     73
     74 000000                                 CIRCX$  LIST
                                        ;
                                        ; CONTEXT AREA OFFSET DEFINITIONS
                                        ;
        000004                                 .ASECT
                    000000                     . = 0
                    000020                     L$NLEN  = 16.           ; MAXIMUM NAME LENGTH
                                        ;
                                        ; The offsets L$NAM to L$PDV inclusive must appear in the same order
                                        ; as in the CIRCX$ and LOCCX$ macros.
                                        ;
        000000                          L$NAM:
        000000                          L$DDM:  .BLKW   1               ; DEVICE NAME
        000002                          L$CTL:  .BLKB   1               ; CONTROLLER NUMBER
        000003                          L$UNT:  .BLKB   1               ; UNIT NUMBER
        000004                          L$PVC:  .BLKW   1               ; 3RD WORD OF PVC NAME
                    000020                      . = 0+L$NLEN    ; SPACE FOR EXPANDED CIRCUIT NAME
        000020                          L$SCN:  .BLKB   L$NLEN          ; CURRENT NAME IN WILDCARD SCAN
        000040                          L$TRB:  .BLKB   1                       ; TRIBUTARY NUMBER
        000041                          L$PFG:  .BLKB   1                       ; PARSE FLAGS
        000042                          L$SLT:  .BLKW   1                       ; Current SLT/PVC address
        000044                          L$NXT:  .BLKW   1                       ; Pointer to next SLT/PVC address
        000046                          L$SNM:  .BLKW   1                       ; REMAINING NUMBER OF SYSTEM LINES
        000050                          L$TPT:  .BLKW   1                       ; Current tributary pointer (-1 for PSI)
        000052                          L$TNM:  .BLKB   1                       ; REMAINING NUMBER OF TRIBUTARIES
        000053                          L$CTB:  .BLKB   1                       ; CURRENT TRIBUTARY NUMBER
        000054                          L$CHN:  .BLKB   1                       ; Channel number / X.25 port number and
        000055                          L$PDV:  .BLKB   1                       ; ... PDV assigned
        000056                          L$MSG:  .BLKW   1                       ; ERROR MESSAGE STRING POINTER
        000060                          L$BUF:  .BLKW   1                       ; SAVED BUFFER POINTER
        000062                          L$OPT:  .BLKB   1                       ; SAVED OPTIONS BYTE
        000063                          L$TYP:  .BLKB   1                       ; LINE-ID FORMAT TYPE
        000064                          L$FLG:  .BLKW   1                       ; FLAG WORD
        000066                          L$SLEN: .BLKB   1                       ; Significant length of circuit name
        000067                          L$PRO:                                  ; Line protocol
        000067                          L$MTYP: .BLKB   1                       ; Network management circuit type
        000070                          L$TSZ:                                  ; Block size for transport circuits
        000070                          L$LTM:                                  ; Listen timer for transport circuits
        000070                          L$FLX:  .BLKW   1                       ; Flags word for X.25 circuit commands
        000072                          L$SCR:  .BLKW   7                       ; SCRATCH BUFFER
```

```
                                                .SBTTL  CHKDEV - CHECK VALIDITY OF DEVICE SPECIFIER
                                        ;+
                                        ; **-CHKDEV-CHECK VALIDITY OF DEVICE SPECIFIER
                                        ;
                                        ; THIS ROUTINE MAKES SURE THAT THE DEVICE SPECIFIED IS KNOWN TO THE SYSTEM,
                                        ; AND THAT THE FIELDS SPECIFIED APPLY.
                                        ;
                                        ; INPUTS:
                                        ;       R5 = ADDRESS OF CONTEXT AREA WITH
                                        ;       L$DDM(R5) = DDM NAME OF LINE IN RAD50
                                        ;       L$CTL(R5) = CONTROLLER NUMBER
                                        ;       L$UNT(R5) = UNIT NUMBER
                                        ;       L$TRB(R5) = TRIBUTARY NUMBER
                                        ;       L$PFG(R5) = PARSE FLAGS
                                        ;
                                        ; OUTPUTS:
                                        ;       IF CC, LINE IS FOUND AND IS LEGAL
                                        ;       ELSE, UNRECOGNIZED LINE OR ILLEGAL LINE-ID (ERROR CODE IN R0)
                                        ;
                                        ; NO REGISTERS MODIFIED
                                        ;-

001036                                  CHKDEV:
                                                .IF DF  S$$BAS ! R$$RTR

                                                CMP     #^RLC ,L$DDM(R5)        ; is it a PAM line card ?
                                                BNE     5$                      ; if NE, no
                                                SEC                             ; assume failure...
                                                MOV     #ME.IID,R0              ; ...invalid id
                                                BITB    #LP$TRB!LP$UNT,L$PFG(R5); trib or unit parsed ?
                                                BNE     100$                    ; if NE, yes - error
                                                MOV     #ME.UCO,R0              ; assume unrecognized component
                                                BITB    #LP$WCN,L$PFG(R5)       ; wildcard controller parsed ?
                                                BEQ     100$                    ; if EQ, not "LC-*"
                                                MOV     #^RPAM,L$DDM(R5)        ; else look like "PAM-*-*"...
                                                BISB    #LP$MUX!LP$UNT!LP$WCN!LP$WUN,L$PFG(R5) ; ...internally

                                                .ENDC   ; DF S$$BAS ! R$$RTR

001036                                  5$:     SWSTK$  100$                    ;; ENTER SYSTEM STATE
                                        ;
                                        ; MAKE SURE THE PROCESS EXISTS, AND IS A DDM
                                        ;
001042                                          CALL    PDVAD                   ;; FIND IT'S PDV ADDRESS
001046  103521                                  BCS     90$                     ;; IF CC, LINE NOT IN SYSTEM

                                                .IF NDF R$$11S

001050  022762  043340  000004                  CMP     #^RKMX,Z.NAM(R2)        ;; IS THIS A KMX?
001056  001515                                  BEQ     90$                     ;; BR IF YES - NOT A VALID CIRCUIT

                                                .ENDC   ; NDF R$$11S
                                        ;
                                        ; MAKE SURE THE LINE EXISTS
                                        ;
001060  017700  000000G                         MOV     @SLTNM,R0               ;; GET NUMBER OF SYSTEM LINES   ; RJK05
001064  017703  000000G                         MOV     @SLTMA,R3               ;; GET ADDRESS OF SYSTEM LINE IN; RJK05
```

```
                                                                 J 10
PRSCIR     CREATED BY  MACRO  ON 29-JUN-85 AT 13:20     PAGE 5

MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME        REFERENCES

CALL               8-131     8-151     8-152     8-154     8-171     8-179     8-183     8-189     8-200     8-208
                   8-226     8-234     8-243    11-381    12-442    13-476    13-477    14-523    14-527    15-629
CIRCX$            #5-61      5-74
MANDF$            #5-61      5-66
PDVDF$            #5-61      5-64
PHBDF$            #5-61      5-67
PVCDF$            #5-61      5-68
RESRG             #5-62      9-283    10-329
RETC              #5-61     14-606
RETURN             8-252     9-284    10-330    11-402    12-459    13-482    14-607    15-637    16-680
SAVRG             #5-62      9-269    10-309
SLTDF$            #5-61      5-65
SOB                8-135     8-159     9-275     9-281    10-322
SWSTK$            13-476    14-523
```

```
                                                                 K 10
**FILE**ID**PRSEXE
```

```
                                                .SBTTL  Parse Area Routing parameters
                                        ;+
                                        ; ** .EAMAX - Parse Maximum Area
                                        ; ** .EACOS - Parse Area Maximum Cost
                                        ; ** .EAHOP - Parse Area Maximum Hops
                                        ;
                                        ; Inputs:       TPARS conventions.
                                        ;
                                        ; Output:       The transition is accepted/rejected.
                                        ;
                                        ;-
                                                .ENABL  LSB

000624                                  .EAMAX::
000624  016767  000000G 000000G                 MOV     .PNUMB,$EAMAX   ; save max areas
000632  001440                                  BEQ     15$             ; don't allow zero
000634  105767  000001G                         TSTB    .PNUMB+1        ; check for > 63.
000640  001035                                  BNE     15$             ; if NE, error
000642  000421                                  BR      10$             ; else join common code

000644                                  .EACOS::
000644  016767  000000G 000000G                 MOV     .PNUMB,$EACOS   ; save area max cost
000652  001430                                  BEQ     15$             ; if EQ, bad value
000654  026727  000000G 000000G                 CMP     .PNUMB,#MAXCST  ; is the value in range ?
000662  101024                                  BHI     15$             ; if HI, error
000664  000410                                  BR      10$             ; else join common code

000666                                  .EAHOP::
000666  016767  000000G 000000G                 MOV     .PNUMB,$EAHOP   ; save area max hops
000674  001417                                  BEQ     15$             ; if EQ, bad value
000676  026727  000000G 000000G                 CMP     .PNUMB,#MAXHOP  ; is the value in range ?
000704  101013                                  BHI     15$             ; if HI, no

000706  005767  000000G                 10$:    TST     .PNUMH          ; 32 bit value
000712  001010                                  BNE     15$             ; if NE, yes - error
000714  032767  000000G 000000G                 BIT     #NF.LV2,$CNFIG  ; are we configured as level 2 router ?
000722  001011                                  BNE     20$             ; if NE, yes
000724  012767  000000G 000000G                 MOV     #ERRPNA,$ERROR  ; else parameter not applicable
000732  000403                                  BR      17$             ; reject transition

000734  012767  000000G 000000G 15$:    MOV     #ERRILN,$ERROR  ; error in number
000742  062716  000002          17$:    ADD     #2,(SP)         ; else reject transition

000746  000207                          20$:    RETURN

                                                .DSABL  LSB

```

Table of contents

```
     5-    53     MACRO LIBRARY CALLS AND LOCAL DATA
     6-    63     BIT DEFINITIONS
     7-    66     PRSLIN - PARSE AN ASCII LINE-ID
     8-   168     CAT5 - CONVERT ASCII TO RAD50
     9-   238     CDTB - CONVERT DECIMAL TO BINARY
    10-   295     CHKMUX - CHECK IF A DEVICE IS A MULTIPLEXOR
    11-   318     CHKDEV - CHECK VALIDITY OF DEVICE SPECIFIER
    12-   390     PDVAD - GET THE PDV ADDRESS
    13-   422     CHKWLD - CHECK FOR WILD CARD FIELD
    390                                                  .SBTTL  PDVAD - GET THE PDV ADDRESS
    391                                          ;+
    392                                          ; **-PDVAD-GET THE PDV ADDRESS
    393                                          ;
    394                                          ; THIS ROUTINE IS CALLED TO RETURN THE PDV INDEX AND ADDRESS
    395                                          ; FOR A DDM.
    396                                          ;
    397                                          ; INPUTS:
    398                                          ;       R5 - ADDRESS OF CONTEXT AREA WITN
    399                                          ;       L$DDM(R5) = DDM NAME IN RAD50
    400                                          ;
    401                                          ; OUTPUTS:
    402                                          ;       IF CC, PROCESS FOUND AND IS A DDM
    403                                          ;          R2 = PDV INDEX
    404                                          ;          R4 = PDV ADDRESS
    405                                          ;       ELSE, ERROR
    406                                          ;-
    407
    408 000662                                   PDVAD:
    409 000662  016502  000000                          MOV     L$DDM(R5),R2            ;; GET PROCESS NAME
    410 000666                                          CALL    @PDVID                  ;; FIND IT'S PDV ID             ; RJK02
    411 000672  103415                                  BCS     10$                     ;; IF CC, LINE NOT IN SYSTEM
    412 000674  010204                                  MOV     R2,R4                   ;; COPY THE PDV INDEX
    413 000676  067704  000000G                         ADD     @PDVTA,R4               ;; GET THE PDV ADDRESS          ; RJK02
    414 000702  011404                                  MOV     (R4),R4                 ;; POINT TO THE PDV
    415 000704  032764  002000  000010                  BIT     #ZF.PSE,Z.FLG(R4)       ;; IS THIS A PSEUDO-SLT?
    416 000712  001004                                  BNE     5$                      ;; BR IF YES - NOT VALID LINE
    417 000714  132764  000001  000010                  BITB    #ZF.DDM,Z.FLG(R4)       ;; IS THE PROCESS A DDM ?
    418 000722  001001                                  BNE     10$                     ;; IF NE, YES
    419 000724  000261                          5$:     SEC                             ;; ELSE, ERROR
    420 000726                                  10$:    RETURN
```

```
                                        .TITLE  PRSLOG - PARSE LOGGING ACTION ROUTINES
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; CFE - PARSE COMMANDS ASSOCIATED WITH  ''DEFINE LINE'' REQUESTS
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  01-JUL-80
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;
```

```
        .TITLE  PRSMOD - PARSE ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - PARSE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-OCT-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                  .SBTTL  .MOMCI - PARSE MAXIMUM CIRCUITS (DTE)
                                        ;+
                                        ; ** .MOMCI - PARSE MAXIMUM CIRCUITS (DTE)
                                        ;
                                        ; INPUTS          .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT          $MCI = PARSED MAXIMUM CIRCUITS
                                        ;                 THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;-

000450                                  .MOMCI::
000450  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
000454  001014                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
000456  012700  000020                          MOV     #16.,R0         ; MUST BE POWER OF 2 (32.-512.)
000462  006300                          10$:    ASL     R0              ; NEXT POWER
000464  020027  001000                          CMP     R0,#512.        ; CHECK UPPER BOUND
000470  003006                                  BGT     ILLNUM          ; IF GT, REJECT TRANSITION
000472  026700  000000G                         CMP     .PNUMB,R0       ; COMPARE WITH NEXT POWER OF 2
000476  003371                                  BGT     10$             ; IF GT, KEEP GOING
000500  010067  000000G                         MOV     R0,$MCI         ; SAVE NEAREST POWER OF 2
000504  000207                                  RETURN

000506  012767  000000G 000000G ILLNUM: MOV     #ERRILN,$ERROR  ; ILLEGAL NUMBER
000514  062716  000002          REJECT: ADD     #2,(SP)         ; REJECT TRANSITION
000520  000207                                  RETURN

```

```
        .TITLE  CFEUPD - UPDATE THE CETAB FILE BEFORE EXITING
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985  BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;         STATE TABLE AND ACTION ROUTINES TO DETERMINE WHICH DEFINITION IT IS.
;
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00    14-DEC-79
;         DECNET-11M/S V3.0
;         DECNET-11M-PLUS V1.0
;
; 3.00    16-APR-82
;         DECNET-11M V3.1
;         DECNET-11M-PLUS V1.1
;
; 4.00    07-NOV-83
;         DECNET-11M V4.0
;         DECNET-11M-PLUS V2.0
;
; 5.00    22-JUL-85
;         DECnet-11M/S V4.2
;         DECnet-11M-Plus V3.0
;         DECnet-Micro/RSX V1.0
;
;

```

```
                                                                    K  2
CFEUPD - UPDATE THE CETAB FILE  MACRO V05.03b Saturday 29-Jun-85 05:15  Page 13-1
Symbol table

ARGLST  000006R         G.TIYR= 000000        $BLANK= 000006        $MOVIT  000436RG      $UPPAR= ****** GX
CLENUP  000570R         G.TSDU= 000036        $BUFI = ****** GX     $MOVPT  000476RG      $UPPSN= ****** GX
DF      000326R     002 G.TSFW= 000024        $BUFR = ****** GX     $MOVST  000446RG      $UPREM= ****** GX
DOLLAR  000322R     002 G.TSGC= 000017        $BYTE = ****** GX     $MVASC= ****** GX     $UPROU= ****** GX
END     000316R     002 G.TSMT= 000022        $CBDMG= ****** GX     $NUMBR= 000002        $UPSLT= ****** GX
ERRFIL= ****** GX       G.TSNL= 000020        $CBOMG= ****** GX     $OBUF   000004RG      $UPSTA= ****** GX
ERROUT= ****** GX       G.TSPC= 000016        $CFERR= ****** GX     $OFDB = ****** GX     $UPSVC= ****** GX
ERRSYN= ****** GX       G.TSPN= 000004        $CFER1= ****** GX     $PSIPT= ****** GX     $UPUNT= ****** GX
FMTSTR  000052R         G.TSPR= 000014        $CFUPD  000170RG      $RAD50= 000016        $WDATA= ****** GX
F.NRBD= ****** GX       G.TSRN= 000010        $CLNUP  000552RG      $RDATA= ****** GX     $ZCDU   000136RG
GINBUF  000156R         G.TSSY= 000034        $CNVD   000526RG      $STRNG= 000004        $ZCDUL= 000007 G
GINDPB  000146R         G.TSTN= 000000        $CNVO   000542RG      $SUBXP= 000010        $ZMFL   000120RG
G$BLEN  000012          G.TSTS= 000032        $DIGIT= 000024        $SVUPD= ****** GX     $ZMFLL= 000007 G
G$CRUI  000004          G.TSVA= 000026        $DNUMB= 000014        $UPBUF= ****** GX     $ZTIM   000127RG
G$LBUI  000002          G.TSVL= 000030        $EDMSG= ****** GX     $UPCNT= ****** GX     $ZTIML= 000007 G
G$LGUI  000010          RX$IAS= 000003        $EDS  = 000012        $UPDDM= ****** GX     $$$FLG= 177777
G$PRUI  000006          RX$11D= 000000        $ERROR  000000RG      $UPDEC= ****** GX     $$$KEY= 000041
G$SYUI  000000          RX$11M= 000001        $EXIT = 000000        $UPDLC= ****** GX     $$$OST= 000020
G.TICP= 000016          RX$11S= 000002        $FAIL = 177777        $UPEVT= ****** GX     $$$STA= 000000
G.TICT= 000014          SEMI  = 000073        $HIVBN= ****** GX     $UPFEA= ****** GX     $$$TMP= 000344R     004
G.TIDA= 000004          SKIPIT  000430R       $IFDB = ****** GX     $UPLLC= ****** GX     .POINT= ****** GX
G.TIHR= 000006          UPDKTB  000000RG  003 $LAMDA= 000000        $UPLOG= ****** GX     .PSTCN= ****** GX
G.TIMI= 000010          UPDSTB  000000RG  002 $LEN    000002RG      $UPNOD= ****** GX     .PSTPT= ****** GX
G.TIMO= 000002          $ALPHA= 000022        $LOVBN= ****** GX     $UPOBJ= ****** GX     .TPARS= ****** GX
G.TISC= 000012          $ANY  = 000020

. ABS.  000012    000   (RW,I,GBL,ABS,OVR)
        000744    001   (RW,I,LCL,REL,CON)
$STATE  000332    002   (RW,D,LCL,REL,CON)
$KTAB   000104    003   (RW,D,LCL,REL,CON)
$KSTR   000347    004   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11544 Words  ( 46 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:41.10
SY:CFEUPD.V2,[132,134]CFEUPD/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]CFEUPD
```

```
                                                     L  2
CFEUPD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:15     PAGE 1
```

```
CFEUSB     CREATED BY  MACRO  ON 29-JUN-85 AT 05:16     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME      REFERENCES
CALLR           #4-61
```

```
**FILE**ID**EVTTBL
```

```
                                                                 K  4
NOSNA - STUB ACTION ROUTINES FO MACRO V05.03b Saturday 29-Jun-85 05:16  Page 4-1
Symbol table

$PUSNA  000004RG        $SNDEF  000000RG        $TMSAC  000000RG        $TMSPU  000000RG        $UPSNA  000000RG
$SESNA  000004RG        $SNLIS  000000RG        $TMSAD  000000RG        $UPSAC  000000RG        $UPSPU  000000RG
$SHSNA  000004RG        $SNPUR  000000RG        $TMSNA  000000RG        $UPSAD  000000RG

. ABS.  000000     000   (RW,I,GBL,ABS,OVR)
        000006     001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 46 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:03.10
SY:NOSNA.V2,[132,134]NOSNA/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]NOSNA
```

```
                                                                 L  4
NOSNA      CREATED BY  MACRO  ON 29-JUN-85 AT 05:16      PAGE 1
```

K 5

```
PRMPTS     CREATED BY  MACRO  ON 29-JUN-85 AT 17:01     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

DBGTP$          #5-90
ISTAT$          #4-63       5-90
MTRAN$          #5-90
PRMPT$          #4-63       5-94      #5-96     #5-97     #5-98     #5-99     #5-100    #5-101    #5-102    #5-103
                #5-104     #5-105     #5-106    #5-107    #5-108    #5-109    #5-110    #5-111    #5-112    #5-113
                #5-114     #5-115     #5-116    #5-117    #5-118    #5-119    #5-120    #5-121    #5-122    #5-123
                #5-124     #5-125     #5-126    #5-127    #5-128    #5-129    #5-130    #5-131    #5-132    #5-133
                #5-134     #5-135     #5-136    #5-137    #5-138    #5-139    #5-140    #5-141    #5-142    #5-143
                #5-144     #5-145     #5-146    #5-147    #5-148    #5-149    #5-150    #5-151    #5-152    #5-153
                #5-154     #5-155     #5-156    #5-157    #5-158    #5-159    #5-160    #5-161    #5-162    #5-163
                #5-164     #5-165     #5-166    #5-167    #5-168    #5-169    #5-170    #5-171    #5-172    #5-173
                #5-174     #5-175     #5-176    #5-177    #5-178    #5-179    #5-180    #5-181    #5-182    #5-183
                #5-184     #5-185     #5-186    #5-187    #5-188    #5-189    #5-190    #5-191    #5-192    #5-193
                #5-194     #5-195     #5-196    #5-197    #5-198    #5-199    #5-200    #5-201    #5-202    #5-203
                #5-204     #5-205     #5-206    #5-207    #5-208    #5-209    #5-210    #5-211    #5-212    #5-213
                #5-214     #5-215     #5-216    #5-217    #5-218    #5-219    #5-220    #5-221    #5-222    #5-223
                #5-224     #5-225     #5-226    #5-227    #5-228    #5-229    #5-230    #5-231    #5-232    #5-233
                #5-234     #5-235     #5-236    #5-237    #5-238    #5-239    #5-240    #5-241    #5-242    #5-243
                #5-244     #5-245     #5-246    #5-247    #5-248    #5-249    #5-250    #5-251    #5-252    #5-253
                #5-254     #5-255     #5-256    #5-257    #5-258    #5-259    #5-260    #5-261    #5-262    #5-263
                #5-264     #5-265     #5-266    #5-267    #5-268    #5-269    #5-270    #5-271    #5-272    #5-273
                #5-274     #5-275     #5-276    #5-277    #5-278    #5-279    #5-280    #5-281    #5-282    #5-283
                #5-284     #5-285     #5-286    #5-287    #5-288    #5-289    #5-290    #5-291    #5-292    #5-293
                #5-294
```

L 5

FILEID**PRMPTO

```
**FILE**ID**PRSACT




PPPPPPPP     RRRRRRRR       SSSSSSSS     AAAAAA       CCCCCCCC   TTTTTTTTTT
PPPPPPPP     RRRRRRRR       SSSSSSSS     AAAAAA       CCCCCCCC   TTTTTTTTTT
PP      PP   RR      RR   SS           AA      AA   CC               TT
PP      PP   RR      RR   SS           AA      AA   CC               TT
PP      PP   RR      RR   SS           AA      AA   CC               TT
PP      PP   RR      RR   SS           AA      AA   CC               TT
PPPPPPPP     RRRRRRRR       SSSSSS     AA      AA   CC               TT
PPPPPPPP     RRRRRRRR       SSSSSS     AA      AA   CC               TT
PP           RR  RR               SS   AAAAAAAAAA   CC               TT
PP           RR  RR               SS   AAAAAAAAAA   CC               TT
PP           RR    RR             SS   AA      AA   CC               TT           ....
PP           RR    RR             SS   AA      AA   CC               TT           ....
PP           RR      RR   SSSSSSSS     AA      AA     CCCCCCCC       TT           ....
PP           RR      RR   SSSSSSSS     AA      AA     CCCCCCCC       TT           ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
    371                                                   .SBTTL  .EXRTM - PARSE EXECUTOR ROUTING TIMER
    372                                         ;+
    373                                         ; INPUTS          .TPARS VARIABLES
    374                                         ;
    375                                         ; OUTPUT          $RTM = PARSED ROUTING TIMER VALUE
    376                                         ;                 THE TRANSITION IS ACCEPTED/REJECTED.
    377                                         ;-
    378
    379 001040                                  .EXRTM::
    380                                         ;;;       TST     .PNUMH            ; SEE IF GT 16 BITS WIDE
    381                                         ;;;       BNE     ILLNUM            ; RECJECT TRANSITION IF IT IS
    382 001040  016767  000000G 000000G                   MOV     .PNUMB,$RTM       ; SAVE IN PARSED DATA AREA
    383 001046  001766                                    BEQ     ILLNUM            ; IF EQ, OUT OF RANGE
    384 001050  000207                                    RETURN
```

```
        000110                          L$ADJ:  .BLKW   1                       ; Pointer to adjacency database
        000112                          L$NOD:  .BLKW   1                       ; Adjacent node address
        000114                          L$COU:  .BLKW   1                       ; Count of adjacency entries
        000116                          L$PLB:  .BLKW   1                       ; PLB address
        000120                          L$PAR:  .BLKW   1                       ; PARAMETER TYPE
        000122                          L$SYL:  .BLKW   1                       ; System line number
        000124                          L$LEN:                                  ; LENGTH OF CONTEXT AREA
        000000                                  .PSECT
                                        ;
                                        ; OFFSETS INTO SCRATCH BUFFER (L$SCR) FOR DMC BASE TABLE COUNTERS
                                        ;
        000124                                  .ASECT
                000000                          .=0
        000000                          B$5:    .BLKB   1                       ; BASE TABLE + 5
        000001                          B$6:    .BLKB   1                       ; BASE TABLE + 6
        000002                          B$7:    .BLKB   1                       ; BASE TABLE + 7
        000003                          B$10:   .BLKB   1                       ; BASE TABLE + 10
        000004                          B$11:   .BLKB   1                       ; BASE TABLE + 11
        000005                          B$12:   .BLKB   1                       ; BASE TABLE + 12
        000006                          B$7S10: .BLKB   1                       ; SUM OF BASE TABLE 7 AND 10
                                                .EVEN
        000000                                  .PSECT
```

```
001070  012301                          10$:    MOV     (R3)+,R1                ;; GET ADDR OF SYSTEM LINE TABLE
001072  126102  000002                          CMPB    L.DDM(R1),R2            ;; DOES THIS PROCESS MATCH ?
001076  001020                                  BNE     20$                     ;; IF NE, NO - KEEP LOOKING
001100  132765  000040  000041                  BITB    #LP$WCN,L$PFG(R5)       ;; WILD CONTROLLER ?
001106  001017                                  BNE     30$                     ;; IF NE YES - SKIP LOOKING FOR SLT
001110  126165  000012  000002                  CMPB    L.CTL(R1),L$CTL(R5)     ;; DOES THIS CONTROLLER NUMBER MATCH ?
001116  001010                                  BNE     20$                     ;; IF NE, NO - KEEP LOOKING
001120  132765  000100  000041                  BITB    #LP$WUN,L$PFG(R5)       ;; WILD UNIT ?
001126  001007                                  BNE     30$                     ;; IF NE, YES - GO ON TO MUX TEST
001130  126165  000013  000003                  CMPB    L.UNT(R1),L$UNT(R5)     ;; DOES THIS UNIT NUMBER MATCH ?
001136  001403                                  BEQ     30$                     ;; IF EQ, FOUND THE CORRECT SLN
001140  005300                          20$:    DEC     R0                      ;; LOOK THROUGH ALL SYSTEM LINES
001142  001352                                  BNE     10$                     ;; ...
001144  000462                                  BR      90$                     ;; IF HERE, LINE NOT FOUND
                                        ;
                                        ; IF TRIBUTARY WAS SPECIFIED, MAKE SURE LINE IS MULTIPOINT MASTER
                                        ;
001146  116103  000003                  30$:    MOVB    L.DLC(R1),R3            ;; GET DLC'S PDV INDEX
001152  067703  000000G                         ADD     @PDVTA,R3               ;; INDEX INTO PDV VECTOR TABLE  ; RJK05
001156  011303                                  MOV     (R3),R3                 ;; GET DLC'S PDV ADDRESS

                                                .IF NDF R$$11S

001160  022763  045452  000004                  CMP     #^RLAB,Z.NAM(R3)        ;; IS IT A PSI LINE?
001166  001451                                  BEQ     90$                     ;; BR IF YES

                                                .ENDC   ; NDF R$$11S

001170  132765  000140  000041                  BITB    #LP$WCN!LP$WUN,L$PFG(R5);; WILD CARD CONTROLLER OR UNIT ?
001176  001055                                  BNE     100$                    ;; IF NE, YES - RETURN (NO SLT TO EXAMINE)
001200  132765  000002  000041                  BITB    #LP$TRB,L$PFG(R5)       ;; WAS A TRIBUTARY SPECIFIED ?
001206  001420                                  BEQ     40$                     ;; IF EQ, NO - RETURN
001210  032761  000020  000000                  BIT     #LF.MTP,L.FLG(R1)       ;; IS THIS A MULTIPOINT CIRCUIT ?
001216  001431                                  BEQ     80$                     ;; IF EQ, NO - ERROR
001220  105761  000014                          TSTB    L.NSTA(R1)              ;; IS THIS A MULTI-POINT MASTER ?
001224  001426                                  BEQ     80$                     ;; IF EQ, NO - ERROR
001226  132765  000200  000041                  BITB    #LP$WTR,L$PFG(R5)       ;; WILD TRIBUTARY ?
001234  001036                                  BNE     100$                    ;; IF NE, YES - SKIP RANGE CHECK
001236  126165  000014  000040                  CMPB    L.NSTA(R1),L$TRB(R5)    ;; ELSE, IS THE TRIBUTARY IN RANGE ?
001244  101010                                  BHI     70$                     ;; IF HI, YES
001246  000415                                  BR      80$                     ;; ELSE ERROR
                                        ;
                                        ; IF NO TRIBUTARY WAS SPECIFIED, MAKE SURE LINE IS NOT MULTIPOINT MASTER
                                        ;
001250                                  40$:
                                                .IF DF  R$$11S

                                                TST     ENTTYP                  ;; PARSING A LINE-ID (NOT CIRCUIT)?
                                                BNE     100$                    ;; IF NE, YES - MISSING TRIB OK

                                                .ENDC   ; NDF R$$11S

001250  032761  000020  000000                  BIT     #LF.MTP,L.FLG(R1)       ;; IS THIS A MULTIPOINT LINE ?
001256  001403                                  BEQ     70$                     ;; IF EQ, NO - VALID CIRCUIT ID
001260  105761  000014                          TSTB    L.NSTA(R1)              ;; IS THIS A MULTIPOINT MASTER ?
001264  001006                                  BNE     80$                     ;; BR IF YES - INVALID CIRCUIT-ID

```

```
**FILE**ID**PRSEXE
```

```
PPPPPPPP     RRRRRRRR       SSSSSSSS   EEEEEEEEEE   XX      XX   EEEEEEEEEE
PPPPPPPP     RRRRRRRR       SSSSSSSS   EEEEEEEEEE   XX      XX   EEEEEEEEEE
PP      PP   RR      RR   SS           EE           XX      XX   EE
PP      PP   RR      RR   SS           EE           XX      XX   EE
PP      PP   RR      RR   SS           EE             XX  XX     EE
PP      PP   RR      RR   SS           EE             XX  XX     EE
PPPPPPPP     RRRRRRRR       SSSSSS     EEEEEEEE         XX       EEEEEEEE
PPPPPPPP     RRRRRRRR       SSSSSS     EEEEEEEE         XX       EEEEEEEE
PP           RR  RR               SS   EE             XX  XX     EE
PP           RR  RR               SS   EE             XX  XX     EE
PP           RR    RR             SS   EE           XX      XX   EE               ....
PP           RR    RR             SS   EE           XX      XX   EE               ....
PP           RR      RR   SSSSSSSS     EEEEEEEEEE   XX      XX   EEEEEEEEEE       ....
PP           RR      RR   SSSSSSSS     EEEEEEEEEE   XX      XX   EEEEEEEEEE       ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
PRSEXE - PARSE EXECUTOR ACTION  MACRO V05.03b Saturday 29-Jun-85 05:18
```

```
                                         .SBTTL  .EXBRT - EXECUTOR BROADCAST ROUTING TIMER
                                 ;+
                                 ; ** .EXBRT- EXECUTOR BROADCAST ROUTING TIMER
                                 ;
                                 ; INPUTS:         TPARS CONVENTIONS.
                                 ;
                                 ; OUTPUT:         $EBRT    = PARSEED BROADCAST ROUTING TIMER
                                 ;-
000750                           .EXBRT::
000750  016767  000000G 000000G          MOV     .PNUMB,$EBRT     ; MOVE IT
000756  001403                           BEQ     10$              ; IF EQ, ERROR
000760  005767  000000G                  TST     .PNUMH           ; TOO HIGH ?
000764  001405                           BEQ     20$              ; IF EQ, OK

000766  012767  000000G 000000G  10$:    MOV     #ERRILN,$ERROR   ; error in number
000774  062716  000002                   ADD     #2,(SP)          ; else reject transition

001000  000207                   20$:    RETURN
```

```
					.TITLE	PRSLIN - PARSE A LINE-ID
					.IDENT	/V05.00/
					.ENABL	LC
				;
				; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
				; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
				;
				;
				; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
				; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
				; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
				; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
				; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
				; TRANSFERRED.
				;
				; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
				; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
				; CORPORATION.
				;
				; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
				; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
				;
				;
				; MODULE DESCRIPTION:
				;
				; NETWORK MANAGEMENT - PARSE A LINE-ID STRING
				;
				;
				; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
				;
				; IDENT HISTORY:
				;
				; 1.00	14-DEC-79
				;	DECNET-11M/S V3.0
				;	DECNET-11M-PLUS V1.0
				;
				; 2.00	16-APR-82
				;	DECNET-11M V3.1
				;	DECNET-11M-PLUS V1.1
				;
				; 4.00	07-NOV-83
				;	DECNET-11M V4.0
				;	DECNET-11M-PLUS V2.0
				;
				; 5.00	22-JUL-85
				;	DECnet-11M/S V4.2
				;	DECnet-11M-Plus V3.0
				;	DECnet-Micro/RSX V1.0
				;
				;

```

```
                                        .SBTTL  CHKWLD - CHECK FOR WILD CARD FIELD
                                ;+
                                ; **-CHKWLD-CHECK FOR WILD CARD FIELD
                                ;
                                ; THIS ROUTINE CHECKS FOR LEGAL WILD CARD SYNTAX.
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS OF NEXT BYTE IN STRING
                                ;       R3 = LEGAL TERMINATING CHARACTER
                                ;       R4 = ADDRESS OF THE END OF THE STRING + 1
                                ;
                                ; OUTPUTS:
                                ;       C-BIT CLEAR:
                                ;       R0 = ADDRESS OF NEXT BYTE IN STRING
                                ;       R2 = COMPLETION CODE ( 0 = MORE IN THE STRING)
                                ;                            ( NOT 0 = NO MORE IN THE STRING)
                                ;
                                ;       C-BIT SET:
                                ;       R2 = COMPLETION CODE ( 0 = '*' NOT FOUND)
                                ;                            ( NOT 0 = '*' FOUND NOT FOLLOWED BY VALID
                                ;                                       TERMINATOR)
                                ;
                                ; REGISTERS:
                                ;       R3, R4, R5 ARE PRESERVED
                                ;-

000730                          CHKWLD:
000730  005002                          CLR     R2                      ; ASSUME '*' NOT FOUND
000732  121027  000052                  CMPB    (R0),#'*                ; WILD CARD ?
000736  001013                          BNE     10$                     ; IF NE, NO - RETURN
000740  005200                          INC     R0                      ; SKIP OVER CHARACTER
000742  005202                          INC     R2                      ; ASSUME END OF STRING
000744  020400                          CMP     R4,R0                   ; AT END OF STRING YET ?
000746  001410                          BEQ     20$                     ; IF EQ, YES
000750  122003                          CMPB    (R0)+,R3                ; LEGAL TERMINATOR ?
000752  001005                          BNE     10$                     ; IF NE, NO - RETURN
000754  020400                          CMP     R4,R0                   ; AT END OF STRING ?
000756  001404                          BEQ     20$                     ; IF EQ, YES - RETURN
000760  103402                          BLO     10$                     ; IF LO, ERROR
000762  005002                          CLR     R2                      ; ELSE, INDICATE MORE IN STRING
000764  000401                          BR      20$                     ; AND RETURN

000766  000261                  10$:    SEC
000770                          20$:    RETURN

                                        .DSABL  LSB

        000001                          .END
        .SBTTL  LOCAL MACROS
;****
; LOCAL MACROS
;****

        .MACRO  PRINT   TEXT
        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
        CALL    $PRINT          ; PRINT MESSAGE
        .ENDM   PRINT

```

```
                                                .SBTTL  .MOCUG - CONVERT USER GROUP TO RAD50
                                        ;+
                                        ; *** - .MOCUG - CONVERT USER GROUP TO RAD50
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT = ADDRESS OF USER GROUP STRING
                                        ;       .PSTCN = LENGTH OF USER GROUP STRING
                                        ;
                                        ; OUTPUTS:
                                        ;       $CUG = 2 WORD RAD50 USER GROUP
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED
                                        ;-

   000000                               .MOCUG::
   000000  005067  000002G                      CLR     $CUG+2          ; CLEAR THE SECOND HALF OF THE USER GROUP
   000004  016700  000000G                      MOV     .PSTPT,R0       ; POINT R0 AT USER GROUP
   000010  005001                               CLR     R1              ; PERIODS NOT ACCEPTABLE
   000012  004767  000000G                      CALL    $CAT5           ; CONVERT USER GROUP TO RAD50
   000016  010167  000000G                      MOV     R1,$CUG         ; STORE THE USER GROUP
   000022  103413                               BCS     10$             ; IF CS, LESS THAN 3 CHARS CONVERTED
   000024  005001                               CLR     R1              ; PERIODS NOT ALLOWED
   000026  004767  000000G                      CALL    $CAT5           ; CONVERT REST OF USER GROUP TO RAD50
   000032  010167  000002G                      MOV     R1,$CUG+2       ; STORE SECOND HALF OF USER GROUP
   000036  022767  000006  000000G              CMP     #6,.PSTCN       ; IS THE USER GROUP TOO BIG ?
   000044  103002                               BHIS    10$             ; IF HIS, NO - OKAY
   000046  062716  000002                       ADD     #2,(SP)         ; ELSE, INDICATE FAILURE
   000052  000207                       10$:    RETURN
```

```
                                                .SBTTL  .MONUM - PARSE GROUP NUMBER
                                        ;+
                                        ; ** .MONUM - PARSE GROUP NUMBER
                                        ;
                                        ; INPUTS        .TPARS VARIABLES
                                        ;
                                        ; OUTPUT        $NUM = PARSED GROUP NUMBER
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;-
000522                                  .MONUM::
000522  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
000526  001367                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
000530  026727  000000G 023417                  CMP     .PNUMB,#9999.   ; CHECK HIGH RANGE
000536  101363                                  BHI     ILLNUM          ; BR IF ILLEGAL NUMBER
000540  016767  000000G 000000G                 MOV     .PNUMB,$NUM     ; MOVE VALUE TO PARSED AREA
000546  001757                                  BEQ     ILLNUM          ; BR IF ZERO
000550  000207                                  RETURN

```

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;
                                ; LOCAL MACROS
                                ;

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                ;
                                ; MACRO CALLS
                                ;

                                        .MCALL  ISTAT$,STATE$,TRAN$,CLOSE$,OPEN$R
                                        .MCALL  DIR$,GTSK$S,GTIM$S


                                ;
                                ; LOCAL DATA
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
        000073                  SEMI    = ';                    ; SEMICOLON FOR TPARS TABLES
000000                          $ERROR::.BLKW   1               ; NON-ZERO IF SYNTAX ERROR
000002                          $LEN::  .BLKW   1               ; LENGTH OF THE OUTPUT BUFFER
000004                          $OBUF:: .BLKW   1               ; CURRENT POSITION IN OUTPUT BUFFER
000006                          ARGLST: .BLKW   18.             ; GTIM$, GTSK$ BUFFER
000052     011     073     040  FMTSTR: .ASCIZ  <11>/; last edit by %2R [%O,%O] on %Y %3Z/
000120     132     106     056  $ZMFL:: .ASCII  /ZF.MFL!/       ; MARK PROCESS FOR LOAD BIT
        000007                          $ZMFLL==.-$ZMFL         ; AND LENGTH
000127     132     106     056  $ZTIM:: .ASCII  /ZF.TIM!/       ; PROCESS NEEDS TIMER SUPPORT
        000007                          $ZTIML==.-$ZTIM         ; AND LENGTH
000136     132     106     056  $ZCOU:: .ASCII  /ZF.COU!/       ; DLC SUPPORTS COUNTERS
        000007                          $ZCOUL==.-$ZCOU
                                        .EVEN
```

```
                                                                  L  2
CFEUPD    CREATED BY  MACRO  ON 29-JUN-85 AT 05:15      PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE              REFERENCES

ARGLST    000006 R         #4-102     13-397     13-398     13-399     13-412
CLENUP    000570 R         #13-387
ERRFIL  = ******   GX       6-214
ERROUT  = ******   GX       6-250
ERRSYN  = ******   GX       6-230
FMTSTR    000052 R         #4-103     13-411
F.NRBD  = ******   GX      *6-221     *6-246     *8-285
GINBUF    000156 R          4-117     #4-131     13-402
GINDPB    000146 R         #4-115     13-400
G$BLEN    000012            4-117     #4-128     4-131
G$CRUI    000004           #4-125
G$LBUI    000002           #4-124
G$LGUI    000010           #4-127     13-402
G$PRUI    000006           #4-126
G$SYUI    000000           #4-123
G.TIYR  = 000000            13-409
G.TSDU  = 000036            13-399
G.TSTN  = 000000            13-397    13-398     13-412
SEMI    = 000073           #4-98
SKIPIT    000430 R         #7-268
UPDKTB    000000 RG        #5-138     6-223
UPDSTB    000000 RG        #5-138     6-225
$ALPHA  = 000022           #5-138
$ANY    = 000020           #5-138
$BLANK  = 000006           #5-138
$BUFI   = ******   GX       6-224     6-243      8-285      9-301      13-414
$BUFR   = ******   GX       6-221     9-300      12-371
$BYTE   = ******   GX       6-210
$CBDMG  = ******   GX       11-348
$CBOMG  = ******   GX       11-353
$CFERR  = ******   GX       6-214     6-230      6-250
$CFER1  = ******   GX       6-243
$CFUPD    000170 RG        #6-204
$CLNUP    000552 RG        #12-369
$CNVD     000526 RG        #11-347
$CNVO     000542 RG        #11-352
$DIGIT  = 000024           #5-138
$DNUMB  = 000014           #5-138
$EDMSG  = ******   GX       13-413
$EOS    = 000012           #5-138
$ERROR    000000 RG        #4-99      *6-205     6-228
$EXIT   = 000000           #5-138
$FAIL   = 177777           #5-138
$GPRM   = ******            5-138
$HIVBN  = ******   GX       6-208
$IFDB   = ******   GX       6-207
$LAMDA  = 000000           #5-138
$LEN      000002 RG        #4-100     *6-219     6-246      *7-269     *12-370    *12-371    *13-415
$LOVBN  = ******   GX       6-209
$MOVIT    000436 RG        #8-284
$MOVPT    000476 RG        #10-322
$MOVST    000446 RG        #9-299
```

```
                                                                  M  2
CFEUPD    CREATED BY  MACRO  ON 29-JUN-85 AT 05:15      PAGE 2
```

```
                                                                              L 3
**FILE**ID**EVTTBL








EEEEEEEEEE   VV      VV   TTTTTTTTTT   TTTTTTTTTT   BBBBBBBB     LL
EEEEEEEEEE   VV      VV   TTTTTTTTTT   TTTTTTTTTT   BBBBBBBB     LL
EE           VV      VV       TT           TT       BB      BB   LL
EE           VV      VV       TT           TT       BB      BB   LL
EE           VV      VV       TT           TT       BB      BB   LL
EE           VV      VV       TT           TT       BB      BB   LL
EEEEEEEE     VV      VV       TT           TT       BBBBBBBB     LL
EEEEEEEE     VV      VV       TT           TT       BBBBBBBB     LL
EE           VV      VV       TT           TT       BB      BB   LL
EE           VV      VV       TT           TT       BB      BB   LL
EE             VV  VV         TT           TT       BB      BB   LL               ....
EE             VV  VV         TT           TT       BB      BB   LL               ....
EEEEEEEEEE       VV           TT           TT       BBBBBBBB     LLLLLLLLLL       ....
EEEEEEEEEE       VV           TT           TT       BBBBBBBB     LLLLLLLLLL       ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLL     SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                              M 3
EVTTBL - EVENT LOGGING TABLES   MACRO V05.03b Saturday 29-Jun-85 05:16
```

```
                                                                  L  4
NOSNA      CREATED BY  MACRO  ON 29-JUN-85 AT 05:16      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

$PUSNA    000004 RG        #4-80
$SESNA    000004 RG        #4-78
$SHSNA    000004 RG        #4-79
$SNDEF    000000 RG        #4-65
$SNLIS    000000 RG        #4-64
$SNPUR    000000 RG        #4-66
$TMSAC    000000 RG        #4-68
$TMSAD    000000 RG        #4-69
$TMSNA    000000 RG        #4-67
$TMSPU    000000 RG        #4-70
$UPSAC    000000 RG        #4-72
$UPSAD    000000 RG        #4-73
$UPSNA    000000 RG        #4-71
$UPSPU    000000 RG        #4-74
```

```
                                                                  M  4
**FILE**ID**PRMPTS
```

L 5

```
**FILE**ID**PRMPTO
```

```
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP     TTTTTTTTTT       000000
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP     TTTTTTTTTT       000000
PP      PP   RR      RR   MMMM  MMMM   PP      PP       TT       00      00
PP      PP   RR      RR   MMMM  MMMM   PP      PP       TT       00      00
PP      PP   RR      RR   MM  MM  MM   PP      PP       TT       00      00
PP      PP   RR      RR   MM  MM  MM   PP      PP       TT       00      00
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP         TT       00      00
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP         TT       00      00
PP           RR  RR       MM      MM   PP               TT       00      00
PP           RR  RR       MM      MM   PP               TT       00      00
PP           RR    RR     MM      MM   PP               TT       00      00       ....
PP           RR    RR     MM      MM   PP               TT       00      00       ....
PP           RR      RR   MM      MM   PP               TT         000000         ....
PP           RR      RR   MM      MM   PP               TT         000000         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

M 5

PRMPTO - PROMPTS, OFFSET ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:02

```
                                                                      L 6
PRSACT - PARSE ACTION ROUTINES  MACRO V05.03b Saturday 29-Jun-85 05:17
Table of contents
```

```
     5-   80    $BL.MAN - MAKE BLANK PROCESSING MANUAL
     6-   98    $BL.AUT - IGNORE BLANK PROCESSING
     6-   99    $BL.A3T - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS
     6-  100    .BL.AU3 - IGNORE BLANKS, ALLOW 3 LETTER ABBREVIATIONS AND FAIL
     7-  121    .HLPAC - HELP ACTION ROUTINE
     8-  137    .NONAM - NODE NAME ACTION ROUTINE
     9-  175    .EXNAM - EXECUTOR NODE NAME ACTION ROUTINE
    10-  199    NODNAM - CONVERT NODE NAME TO RAD50
    11-  230    .RNINI - INITIALIZE NODE ADDRESS
    11-  244    .RNAD2 - SECOND PART OF NODE ADDRESS
    11-  288    .RNADD - STORE NODE ADDRESS
    11-  289    .RNAD3 - CHECK PARSED NODE ADDRESS
    12-  341    .EXADD - STORE EXECUTOR NODE ADDRESS
    13-  371    .EXRTM - PARSE EXECUTOR ROUTING TIMER
    14-  386    .RNNAM - NODE NAME IDENTIFICATION
    15-  428    .RNHNA  - PARSE REMOTE OR EXE NODE HOST
    16-  455    FNDNAM - FIND NODE NAME
    17-  483    .PRMDA - PARSE MAXIMUM DATA
    18-  510    .PRMWI - PARSE MAXIMUM WINDOW
```

```
                                                .SBTTL  .RNNAM - NODE NAME IDENTIFICATION
                                        ;+
                                        ; *** - .RNNAM - NODE NAME IDENTIFICATION
                                        ;
                                        ; INPUT:
                                        ;       .PSTPT = NODE NAME STRING POINTER
                                        ;       .PSTCT = NODE STRING LENGTH
                                        ;
                                        ; OUTPUT:
                                        ;       $NODAD = THE NODE ADDRESS THAT CORRESPONDS TO THE NODE NAME
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-

001052                                  .RNNAM::
001052  022767  000000G 000000G                 CMP     #ERRILC,$ERROR  ; ERROR DETECTED IN NODE ADDR ACTION ROUTINE ?
001060  001041                                  BNE     40$             ; IF NE, YES - PRESERVE ERROR CODE & GET OUT
001062  004767  177306                          CALL    NODNAM          ; TEMPORARILY SAVE THE NODE NAME
001066  103436                                  BCS     40$             ; IF CS, ERROR

001070  026767  000000G 000000G                 CMP     $NODNA,$NTNAM   ; IS THIS THE EXECUTOR NODE NAME ?
001076  001014                                  BNE     20$             ; IF NE, NO
001100  026767  000002G 000002G                 CMP     $NODNA+2,$NTNAM+2 ; IS IT REALLY THE EXECUTOR NODE ?
001106  001010                                  BNE     20$             ; IF NE, NO
001110  022767  000000G 000000G                 CMP     #CM.LIS,$CMAND  ; IS THIS A LIST COMMAND ?
001116  001013                                  BNE     25$             ; IF NE, NO .. ERROR
001120  012767  000000G 000000G                 MOV     #CO.EXE,$COMP   ; ELSE, SET FOR EXECUTOR NODE
001126  000420                                  BR      50$             ; AND LEAVE
001130  004767  000120                  20$:    CALL    FNDNAM          ; FIND NODE NAME IN TEMPLATES
001134  103410                                  BCS     30$             ; IF CS, NOT FOUND
                                        ;
                                        ; FOUND RIGHT TEMPLATE
                                        ;
001136  016067  000010  000000G                 MOV     C.ADD(R0),$NODAD ; SAVE NODE ADDRESS
001144  000411                                  BR      50$             ; AND LEAVE
001146  012767  000105' 000000G 25$:    MOV     #ERREXE,$ERROR  ; INVALID USE OF EXECUTOR NODE NAME
001154  000403                                  BR      40$             ; AND REJECT
001156  012767  000000G 000000G 30$:    MOV     #ERRNOD,$ERROR  ; NODE NOT IN CONFIGURATION
001164  062716  000002                  40$:    ADD     #2,(SP)         ; REJECT THE TRANSITION
001170  000207                          50$:    RETURN
```

```
                                                   .SBTTL  BIT DEFINITIONS
                                          ;
                                          ; PARSE FLAG DEFINITIONS (L$PFG)
                                          ;
                 000001                            LP$UNT = 1                          ; UNIT NUMBER FOUND
                 000002                            LP$TRB = 2                          ; TRIBUTARY NUMBER FOUND
                 000004                            LP$MUX = 4                          ; DEVICE IS MUX
                 000010                            LP$MPT = 10                         ; LINE IS MULTIPOINT
                 000020                            LP$WDV = 20                         ; WILD CARD DEVICE NAME FOUND
                 000040                            LP$WCN = 40                         ; WILD CARD CONTROLLER NUMBER FOUND
                 000100                            LP$WUN = 100                        ; WILD CARD UNIT NUMBER FOUND
                 000200                            LP$WTR = 200                        ; WILD CARD TRIBUTARY NUMBER FOUND
                 000360                            LP$WLD = LP$WDV!LP$WCN!LP$WUN!LP$WTR ; WILD CARD FIELD MASK
                                          ;
                                          ; FLAGS WORD BIT DEFINITIONS (L$FLG)
                                          ;
                 000001                            LF$REA = 1                          ; READ COUNTERS OPERATION
                 000002                            LF$ZER = 2                          ; ZERO COUNTERS OPERATION
                 000004                            LF$SKP = 4                          ; SKIP NEXT "FIND NEXT LINE" OPERATION.
                                                                                       ; THIS IS USED TO FORCE AN EXTRA PASS
                                                                                       ; FOR A MULTIPOINT LINE TO RETURN THE
                                                                                       ; CONTROLLER COUNTERS AS WELL AS ALL
                                                                                       ; OF THE TRIBUTARY COUNTERS.
                 000010                            LF$VR2 = 10                         ; CONNECTED TO VERSION 2.0 NCP
                 000020                            LF$END = 20                         ; EXECUTOR IS AN ENDNODE
                 000040                            LF$SIG = 40                         ; SHOWING SIGNIFIGANT LINES/CIRCUITS
                 040000                            LF$MLT = 40000                      ; MULTIPLE ADJACENCY FLAG
                 100000                            LF$SEG  = 100000                    ; SEGMENTED RESPONSE IN PROGRESS
                                          ;
                                          ; BIT POSITION DEFINITIONS
                                          ;
                 000001                            BIT0 = 1
                 000002                            BIT1 = 2
                 000004                            BIT2 = 4
                 000010                            BIT3 = 10
                 000020                            BIT4 = 20
                 000040                            BIT5 = 40
                 000100                            BIT6 = 100
                 000200                            BIT7 = 200
                 000400                            BIT8 = 400
                 001000                            BIT9 = 1000
                 002000                            BIT10 = 2000
                 004000                            BIT11 = 4000
                 010000                            BIT12 = 10000
                 020000                            BIT13 = 20000
                 040000                            BIT14 = 40000
                 100000                            BIT15 = 100000
      75
```

CHKDEV - CHECK VALIDITY OF DEVICE SPECIFIER

```
001266  017765  000000G 000122  70$:    MOV     @SLTNM,L$SYL(R5)        ;; GET # OF SYSTEM LINES        ; RJK05
001274  160065  000122                  SUB     R0,L$SYL(R5)            ;; COMPUTE SYSTEM LINE NUMBER
001300  000414                          BR      100$                    ;; RETURN

001302  012766  177767  000000G 80$:    MOV     #ME.IID,R$R0(SP)        ;; SET USER R0 TO INVALID CIRCUIT-ID
001310  000403                          BR      95$

001312  012766  177770  000000G 90$:    MOV     #ME.UCO,R$R0(SP)        ;; SET USER R0 TO UNRECOGNIZED CIRCUIT
001320                          95$:    RETC    R0                      ;; SET/CLEAR USER CARRY
001332                          100$:   RETURN                          ;; AND RETURN TO USER MODE
```

Table of contents

```
     5-   69    .EXID - STORE EXECUTOR ID - QUOTED
     5-   70    .EXID1 - STORE EXECUTOR ID NON QUOTED
     6-  125    .EXLNK - SET MAXIMUM NUMBER OF LOGICAL LINKS
     7-  146    .EXHOP - SET MAXIMUM NUMBER OF HOPS IN NETWORK
     8-  168    .EXCOS - SET MAXIMUM LINE COST IN NETWORK
     9-  190    .EXNN - SET MAXIMUM NODE NUMBER IN NETWORK
    10-  225    .EXNC - SET NUMBER OF NODE COUNTERS
    11-  259    .EXHOS - STORE HOST NODE ADDRESS
    12-  290    .EXSUB  - PARSE EXECUTOR SUBADDRESS
    12-  291    .EXSB2 - PARSE EXECUTOR SUBADDRESS RANGE
    13-  326    Parse Area Routing parameters
    14-  374    .EXBRT - EXECUTOR BROADCAST ROUTING TIMER
    15-  393    .EXBEA - EXECUTOR MAXIMUM BROADCAST NONROUTERS
    16-  414    .EXSEG - EXECUTOR SEGMENT BUFFER SIZE
    17-  433    .EXINA - EXECUTOR INACTIVITY TIMER
    18-  454    .EXINT - EXECUTOR INCOMING TIMER
    19-  475    .EXOUT - EXECUTOR OUTGOING TIMER
    20-  496    .EXRET - EXECUTOR RETRANSMIT FACTOR
```

```
                                                .SBTTL  .EXBEA - EXECUTOR MAXIMUM BROADCAST NONROUTERS
                                        ;+
                                        ; ** .EXBEA - EXECUTOR MAXIMUM BROADCAST NONROUTERS
                                        ;
                                        ; INPUTS:         TPARS CONVENTIONS.
                                        ;
                                        ; OUTPUT:         $EBEA    = PARSED EXECUTOR MAXIMUM BROADCAST NONROUTERS
                                        ;-
001002                                  .EXBEA::
001002  016767  000000G 000000G                 MOV     .PNUMB,$EBEA    ; MOVE IT
001010  001407                                  BEQ     10$             ; IF EQ, ERROR
001012  026727  000000G 001776                  CMP     .PNUMB,#1022.   ; TOO HIGH ?
001020  101003                                  BHI     10$             ; YES
001022  005767  000000G                         TST     .PNUMH          ; TOO HIGH ?
001026  001405                                  BEQ     20$             ; IF EQ, OK

001030  012767  000000G 000000G 10$:            MOV     #ERRILN,$ERROR  ; error in number
001036  062716  000002                          ADD     #2,(SP)         ; else reject transition

001042  000207                          20$:    RETURN
```

```
                                                .SBTTL  MACRO LIBRARY CALLS AND LOCAL DATA
                                        ;
                                        ; MACRO LIBRARY CALLS
                                        ;
                                                .MCALL  PDVDF$,SLTDF$,MANDF$,RETC,CIRCX$

   000000                                       PDVDF$                          ; DEFINE PDV OFFSETS
   000000                                       SLTDF$                          ; DEFINE SLT OFFSETS
   000000                                       MANDF$                          ; NICE PROTOCOL SYMBOLS

   000000                                       CIRCX$  LIST
                                        ;
                                        ; CONTEXT AREA OFFSET DEFINITIONS
                                        ;
   000004                                       .ASECT
                 000000                         . = 0
                 000020                         L$NLEN  = 16.           ; MAXIMUM NAME LENGTH
                                        ;
                                        ; The offsets L$NAM to L$PDV inclusive must appear in the same order
                                        ; as in the CIRCX$ and LOCCX$ macros.
                                        ;
   000000                               L$NAM:
   000000                               L$DDM:  .BLKW   1               ; DEVICE NAME
   000002                               L$CTL:  .BLKB   1               ; CONTROLLER NUMBER
   000003                               L$UNT:  .BLKB   1               ; UNIT NUMBER
   000004                               L$PVC:  .BLKW   1               ; 3RD WORD OF PVC NAME
                 000020                         . = 0+L$NLEN    ; SPACE FOR EXPANDED CIRCUIT NAME
   000020                               L$SCN:  .BLKB   L$NLEN          ; CURRENT NAME IN WILDCARD SCAN
   000040                               L$TRB:  .BLKB   1                       ; TRIBUTARY NUMBER
   000041                               L$PFG:  .BLKB   1                       ; PARSE FLAGS
   000042                               L$SLT:  .BLKW   1                       ; Current SLT/PVC address
   000044                               L$NXT:  .BLKW   1                       ; Pointer to next SLT/PVC address
   000046                               L$SNM:  .BLKW   1                       ; REMAINING NUMBER OF SYSTEM LINES
   000050                               L$TPT:  .BLKW   1                       ; Current tributary pointer (-1 for PSI)
   000052                               L$TNM:  .BLKB   1                       ; REMAINING NUMBER OF TRIBUTARIES
   000053                               L$CTB:  .BLKB   1                       ; CURRENT TRIBUTARY NUMBER
   000054                               L$CHN:  .BLKB   1                       ; Channel number / X.25 port number and
   000055                               L$PDV:  .BLKB   1                       ; ... PDV assigned
   000056                               L$MSG:  .BLKW   1                       ; ERROR MESSAGE STRING POINTER
   000060                               L$BUF:  .BLKW   1                       ; SAVED BUFFER POINTER
   000062                               L$OPT:  .BLKB   1                       ; SAVED OPTIONS BYTE
   000063                               L$TYP:  .BLKB   1                       ; LINE-ID FORMAT TYPE
   000064                               L$FLG:  .BLKW   1                       ; FLAG WORD
   000066                               L$SLEN: .BLKB   1                       ; Significant length of circuit name
   000067                               L$PRO:                                  ; Line protocol
   000067                               L$MTYP: .BLKB   1                       ; Network management circuit type
   000070                               L$TSZ:                                  ; Block size for transport circuits
   000070                               L$LTM:                                  ; Listen timer for transport circuits
   000070                               L$FLX:  .BLKW   1                       ; Flags word for X.25 circuit commands
   000072                               L$SCR:  .BLKW   7                       ; SCRATCH BUFFER
   000110                               L$ADJ:  .BLKW   1                       ; Pointer to adjacency database
   000112                               L$NOD:  .BLKW   1                       ; Adjacent node address
   000114                               L$COU:  .BLKW   1                       ; Count of adjacency entries
   000116                               L$PLB:  .BLKW   1                       ; PLB address
   000120                               L$PAR:  .BLKW   1                       ; PARAMETER TYPE
   000122                               L$SYL:  .BLKW   1                       ; System line number
   000124                               L$LEN:                                  ; LENGTH OF CONTEXT AREA
```

```
Symbol table

A$$CHK= 000000     LF$MLT= 040000     L$MTYP  000067     MC$CAP  001440     MC$SYS  000024
A$$CPS= 000000     LF$REA= 000001     L$NAM   000000     MC$CCL  001445     MC$SYZ  000000
A$$PRI= 000000     LF$SEG= 100000     L$NLEN= 000020     MC$CDC  002045     MC$TBR  001140
A$$TRP= 000000     LF$SIG= 000040     L$NOD   000112     MC$CDP  001441     MC$TBS  001141
BIT0  = 000001     LF$SKP= 000004     L$NXT   000044     MC$CIF  001465     MC$TDR  002053
BIT1  = 000002     LF$VR2= 000010     L$OPT   000062     MC$CLD  001464     MC$TIM  000000
BIT10 = 002000     LF$ZER= 000002     L$PAR   000120     MC$COU  100000     MC$TYP  007777
BIT11 = 004000     LF.ACT= 100000     L$PDV   000055     MC$CTL  001454     MC$UBU  002052
BIT12 = 010000     LF.BRO= 000400     L$PFG   000041     MC$CTR  001452     MC$UFD  002047
BIT13 = 020000     LF.BWT= 000007     L$PLB   000116     MC$CTS  001453     MC$UMR  001132
BIT14 = 040000     LF.ENA= 002000     L$PRO   000067     MC$DOV  002050     MC$UMS  001133
BIT15 = 100000     LF.LPB= 001000     L$PVC   000004     MC$LBR  001750     MC$VAX  000003
BIT2  = 000004     LF.MDC= 000100     L$SCN   000020     MC$LBS  001751     MC$WID  060000
BIT3  = 000010     LF.MFL= 004000     L$SCR   000072     MC$LDI  001774     MC$WIH  040000
BIT4  = 000020     LF.MTP= 000020     L$SLEN  000066     MC$LDO  001775     MC$WIL  020000
BIT5  = 000040     LF.PAC= 000200     L$SLT   000042     MC$LDR  001762     MC$W08  020000
BIT6  = 000100     LF.RDY= 040000     L$SNM   000046     MC$LDS  001763     MC$W16  040000
BIT7  = 000200     LF.REA= 010000     L$SYL   000122     MC$LLB  002021     MC$W32  060000
BIT8  = 000400     LF.SER= 000040     L$TNM   000052     MC$LLP  002115     MC$XBR  001750
BIT9  = 001000     LF.TIM= 000010     L$TPT   000050     MC$LLR  002007     MC$XBS  001751
B$10    000003     LF.UNL= 020000     L$TRB   000040     MC$LOO  000001     MC$XCJ  000322
B$11    000004     LF.X2P= 000000     L$TSZ   000070     MC$LRB  002020     MC$XCR  002260
B$12    000005     LN.CLO= 000000     L$TYP   000063     MC$LRP  002114     MC$XCS  002261
B$5     000000     LN.DUM= 000005     L$UNT   000003     MC$LRR  002006     MC$XDR  001762
B$6     000001     LN.LOA= 000004     L$$ASG= 000000     MC$LSI  002032     MC$XDS  001763
B$7     000002     LN.LOO= 000003     L$$DRV= 000000     MC$LST  002033     MC$XFR  002272
B$7S10  000006     LN.OAU= 000003     L$$P11= 000001     MC$LUP  006072     MC$XFS  002273
CAT5    000256R    LN.OFF= 000001     L$$11R= 000000     MC$MAP  010000     MC$XLJ  000323
CDTB    000400R    LN.ON = 000000     L.COST  000015     MC$MBL  001764     MC$XLR  002330
CHKDEV  000530R    LN.OOP= 000004     L.CTL   000012     MC$MBX  004420     MC$XMA  000310
CHKMUX  000476R    LN.OPE= 000001     L.CVA   177776     MC$MBY  001752     MC$XMC  002305
CHKWLD  000730R    LN.REF= 000002     L.DDM   000002     MC$MPX  004406     MC$XMS  002304
CVTBL   000250R    LN.SER= 000002     L.DDS   000004     MC$NAP  001604     MC$XNR  002332
C$$CKP= 000000     LN.STA= 000017     L.DLC   000003     MC$NBR  001130     MC$XRC  002316
C$$ORE= 000400     LN.SUB= 000360     L.DLM   000006     MC$NBS  001131     MC$XRR  002331
C$$RSH= 177564     LN.TRI= 000006     L.DLS   000010     MC$NCR  001154     MC$XRS  002342
D$$BUG= 177514     LP$MPT= 000010     L.FLG   000000     MC$NCS  001155     MC$020  000002
D$$ISK= 000000     LP$MUX= 000004     L.KRBA  000016     MC$NML  001274     MD$CI   000007
D$$L11= 000001     LP$TRB= 000002     L.LEN = 000022     MC$NMR  001142     MD$CNA  000003
D$$YNC= 000000     LP$UNT= 000001     L.MPF   000022     MC$NMS  001143     MD$COU  000001
D$$YNM= 000000     LP$WCN= 000040     L.NMST  000020     MC$NND  001606     MD$DA   000010
E$$XPR= 000000     LP$WDV= 000020     L.NSTA  000014     MC$NNU  001605     MD$DL   000004
F$$LVL= 000001     LP$WLD= 000360     L.OWNR  000021     MC$NOP  001607     MD$DLV  000020
G$$TPP= 000000     LP$WTR= 000200     L.UNT   000013     MC$NOR  000000     MD$DMC  000014
G$$TSS= 000000     LP$WUN= 000100     MB$DIS  000001     MC$NPF  001616     MD$DMF  000046
G$$TTK= 000000     L$ADJ   000110     MB$ENA  000000     MC$NPR  001630     MD$DMP  000022
G$$WRD= 000000     L$BUF   000060     MB$FUL  000002     MC$NRE  001200     MD$DMR  000050
I$$RAR= 000000     L$CHN   000054     MB$MIX  000002     MC$NRT  001166     MD$DMV  000042
I$$RDN= 000000     L$COU   000114     MB$ONE  000001     MC$NVR  001642     MD$DN   000016
K$$CNT= 177546     L$CTB   000053     MB$RXO  000001     MC$P08  000000     MD$DP   000000
K$$CSR= 177546     L$CTL   000002     MB$TXO  000000     MC$P11  000001     MD$DPV  000044
K$$LDC= 000000     L$DDM   000000     MB$ZER  000000     MC$RFL  002046     MD$DQ   000006
K$$TPS= 000074     L$FLG   000064     MC$BAB  006073     MC$SBU  002051     MD$DTE  000024
LC$EXT  000000     L$FLX   000070     MC$BID  001765     MC$SFL  002044     MD$DU   000002
LC$INT  000001     L$LEN   000124     MC$BMC  001767     MC$SYC  000012     MD$DUP  000012
LD$LP = 000000     L$LTM   000070     MC$BSC  001766     MC$SYL  000036     MD$DV   000026
LF$END= 000020     L$MSG   000056     MC$CAC  001442     MC$SYR  000050     MD$DZ   000030
```

```
                                                 .SBTTL  CKEVE  - COMMON ROUTINE TO CHECK AND PROCESS AN EVENT NUMBER
                                        ;+
                                        ;
                                        ; ***  CKEVE  - COMMON ROUTINE TO CHECK AND PROCESS AN EVENT NUMBER
                                        ;
                                        ; FUNCTION:
                                        ;
                                        ;   CHECK RANGE OF EVENT NUMBER AND IF OK, SET PROPER BIT IN EVENT MASK.
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;   R0      - EVENT NUMBER
                                        ;
                                        ; OUTPUTS:
                                        ;
                                        ;   C-BIT   - IF SUCCESS, CLEAR;  ELSE SET
                                        ;   R0      - IF SUCCESS, PRESERVED;  ELSE -1
                                        ;   $EVENT  - BIT SET TO CORRESPOND TO EVENT NUMBER
                                        ;
                                        ; REGISTERS R3, R4, R5 ARE PRESERVED
                                        ;-
                                        ;
                                        ; PROCEDURE BODY
                                        ;
 000000                                 CKEVE:
 000000  005700                                  TST     R0                       ;CHECK NUMBER
 000002  002426                                  BLT     30$                      ;BR IF ERROR
 000004  020027  000077                          CMP     R0,#63.                  ;COMPARE NUMBER TO 63.
 000010  003023                                  BGT     30$                      ;BR IF <63.
 000012  010046                                  MOV     R0,-(SP)                 ;SAVE THE NUMBER
 000014  010001                                  MOV     R0,R1                    ;COPY THE NUMBER
 000016  006200                                  ASR     R0                       ;MAKE THE NUMBER A BYTE INDEX
 000020  006200                                  ASR     R0                       ;BY DIVIDING IT
 000022  006200                                  ASR     R0                       ;BY EIGHT
 000024  042701  177770                          BIC     #177770,R1               ;FIGURE BIT POSITION WITHIN BYTE
 000030  012702  000001                          MOV     #1,R2                    ;START WITH A BIT IN POSITION 0
 000034  005701                         10$:     TST     R1                       ;CHECK SHIFT COUNT
 000036  003403                                  BLE     20$                      ;BR IF SHIFT COUNT <= 0
 000040  006302                                  ASL     R2                       ;SHIFT BIT LEFT ONE POSITION
 000042  005301                                  DEC     R1                       ;DECREMENT SHIFT COUNT
 000044  000773                                  BR      10$
 000046  150260  000000G                20$:     BISB    R2,$EVENT(R0)            ;MERGE BIT INTO BIT ACCUMULATOR
 000052  012600                                  MOV     (SP)+,R0                 ;RESTORE THE NUMBER
 000054  000241                                  CLC                              ;RETURN SUCCESS CONDITION CODE
 000056  000401                                  BR      40$
 000060  000261                         30$:     SEC                              ; INDICATE ERROR
 000062  000207                         40$:     RETURN
```

```
                                          .SBTTL  .MOCAT - PARSE CALL TIMER
                                  ;+
                                  ; ** .MOCAT - PARSE CALL TIMER
                                  ;
                                  ; INPUTS         .TPARS VARIABLES.
                                  ;
                                  ; OUTPUT         $CALT = PARSED CALL TIMER
                                  ;                THE TRANSITION IS ACCEPTED/REJECTED.
                                  ;-
000054                            .MOCAT::
000054  005767  000000G                   TST     .PNUMH           ; HIGH ORDER WORD
000060  001007                            BNE     10$              ; REJECT IF TOO HIGH
000062  105767  000001G                   TSTB    .PNUMB+1         ; TEST HIGH BYTE
000066  001004                            BNE     10$              ; BR IF  ILLEGAL NUMBER
000070  116767  000000G 000000G           MOVB    .PNUMB,$CALT     ; MOVE VALUE TO PARSED AREA
000076  000207                            RETURN
000100  000167  000402            10$:    JMP     ILLNUM
```

```
                                        .SBTTL  .MOINC  - INITIALIZE CHANNEL LIST
                                ;+
                                ;** .MOINC - INITIALIZE CHANNEL LIST
                                ;
                                ; INPUTS        NONE.
                                ;
                                ; OUTPUT        THE CHANNEL LIST IS INITIALIZED.
                                ;
                                ;-

000552                          .MOINC::
000552  005067  000000G                 CLR     $CHNLS          ; NO CHANNELS, YET
000556  000207                          RETURN

```

```
                                                                         M  1
CFEUPD - UPDATE THE CETAB FILE  MACRO V05.03b Saturday 29-Jun-85 05:15  Page 4-1
MACRO DEFINITIONS

    111                                         .LIST   BEX
    112                                 ;
    113                                 ; DPB FOR GIN$ DIRECTIVE
    114                                 ;
    115 000146     251      004         GINDPB: .BYTE   169.,4.
    116 000150  000001                          .WORD   1
    117 000152  000156' 000005                  .WORD   GINBUF,<G$BLEN/2>
    118                                 ;
    119                                 ; OFFSETS FOR RETURNED INFO:
    120                                 ;
    121 000000                                  .ASECT
    122         000000                          .=0
    123 000000                          G$SYUIC: .BLKW  1               ; SYSTEM UIC
    124 000002                          G$LBUIC: .BLKW  1               ; LIBRARY UIC
    125 000004                          G$CRUIC: .BLKW  1               ; CURRENT UIC
    126 000006                          G$PRUIC: .BLKW  1               ; PROTECTION UIC
    127 000010                          G$LGUIC: .BLKW  1               ; LOGIN UIC
    128 000012                          G$BLEN:
    129 000156                                  .PSECT
    130
    131 000156                          GINBUF: .BLKB   G$BLEN
    132
```

```
                                                                         N  1
CFEUPD - UPDATE THE CETAB FILE  MACRO V05.03b Saturday 29-Jun-85 05:15  Page 5
CFEUPD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:15     PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

$MVASC   = ******  GX        9-305      10-327
$NUMBR   = 000002           #5-138
$OBUF      000004 RG        #4-101     *9-306      10-326     *10-328     12-370
$OFDB    = ******  GX        6-220     *6-246     *8-285
$PSIPT   = ******  GX       13-388
$RAD50   = 000016           #5-138
$RDATA   = ******  GX        6-217
$RONLY   = ******            5-138      5-138      5-138
$STRNG   = 000004           #5-138
$SUBXP   = 000010           #5-138
$SVUPD   = ******  GX       13-391
$UPEVT   = ******  GX       13-395
$UPOBJ   = ******  GX       13-394
$UPPSN   = ******  GX       13-390
$UPREM   = ******  GX       13-396
$UPSVC   = ******  GX       13-393
$WDATA   = ******  GX        6-248
$ZCOU      000136 RG        #4-108      4-109
$ZCOUL   = 000007  G        #4-109
$ZMFL      000120 RG        #4-104      4-105
$ZMFLL   = 000007  G        #4-105
$ZTIM      000127 RG        #4-106      4-107
$ZTIML   = 000007  G        #4-107
$$$FLG   = 177777           #5-138
$$$KEY   = 177777           #5-138
.POINT   = ******  GX        6-211
.PSTCN   = ******  GX       10-323
.PSTPT   = ******  GX       10-325
.TPARS   = ******  GX        6-226
```

Table of contents

```
    3-   48     Macro calls
    3-   73     Known events table
```

```
                                                                              M 4
**FILE**ID**PRMPTS
```

```
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP     TTTTTTTTTT     SSSSSSSS
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP     TTTTTTTTTT     SSSSSSSS
PP      PP   RR      RR   MMMM  MMMM   PP      PP       TT       SS
PP      PP   RR      RR   MMMM  MMMM   PP      PP       TT       SS
PP      PP   RR      RR   MM  MM  MM   PP      PP       TT       SS
PP      PP   RR      RR   MM  MM  MM   PP      PP       TT       SS
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP         TT         SSSSSS
PPPPPPPP     RRRRRRRR     MM      MM   PPPPPPPP         TT         SSSSSS
PP           RR  RR       MM      MM   PP               TT               SS
PP           RR  RR       MM      MM   PP               TT               SS
PP           RR    RR     MM      MM   PP               TT               SS        ....
PP           RR    RR     MM      MM   PP               TT               SS        ....
PP           RR      RR   MM      MM   PP               TT       SSSSSSSS          ....
PP           RR      RR   MM      MM   PP               TT       SSSSSSSS          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                              N 4
PRMPTS - PROMPTS. STRING ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:01
```

Table of contents

```
     5-    59     MACROS AND SYMBOLS AND INITIALIZATION
     6-    73     PROMPTS
```

```
                        .TITLE  PRSACT - PARSE ACTION ROUTINES
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1984, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - PARSE ACTION ROUTINES
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                                 .SBTTL  .RNHNA  - PARSE REMOTE OR EXE NODE HOST
                                         ;+
                                         ; INPUTS:       tpars variables
                                         ;
                                         ; OUTPUT:       $NODAD  = HOST NODE ADDRESS
                                         ;
                                         ;       TRANSITION REJECTED IF NODE NAME IS NOT EXEC'S AND IT IS NOT FOUND
                                         ;       IN REMOTE NODE LIST
                                         ;-

 001172                                  .RNHNA::
 001172  004767  177176                          CALL    NODNAM          ; CONVERT NAME TO RAD50
 001176  103423                                  BCS     20$             ; IF CS, BAD NAME, REJECT TRANSISTION
 001200  026767  000000G 000000G                 CMP     $NODNA,$NTNAM   ; EXECUTOR NAME ?
 001206  001010                                  BNE     10$             ; IF NE, NO
 001210  026767  000002G 000002G                 CMP     $NODNA+2,$NTNAM+2 ; EXEC NAME  ?
 001216  001004                                  BNE     10$             ; IF NE, NO
 001220  016767  000000G 000000G                 MOV     $NTADD,$NODAD   ; ELSE, TAKE EXECUTOR NODE ADDRESS
 001226  000411                                  BR      30$             ; AND MAKE LIKE WE HAD IT ALL ALONG

 001230  004767  000020          10$:            CALL    FNDNAM          ; TAKE NAME AND GET REMOTE ADDRESS
 001234  103404                                  BCS     20$             ; IF CS, NO ADDRESS, REJECT TRANSITION
 001236  016067  000010  000000G                 MOV     C.ADD(R0),$NODAD; TAKE ADDRESS FROM TEMPLATE
 001244  000402                                  BR      30$             ; AND ACCEPT TRANSITION
 001246  062716  000002          20$:            ADD     #2,(SP)
 001252  000207                  30$:            RETURN
```

```
     77                                                 .SBTTL  LOCAL DATA
     78                                         ;
     79                                         ; LOCAL DATA
     80                                         ;
     81                                           .IF DF R$$11S
     82                                         ENTITYP: .WORD   0                            ; ENTITY TYPE
     83                                           .IFF
     84 000000       130      062      065     XPREF:  .ASCII  /X25-/                       ; Prefix for new X.25 PVC's
        000003       055
     85              000004                             XPREFL = .-XPREF
     86                                           .ENDC ; DF R$$11S
     87                                                 .EVEN
```

```
                                        .SBTTL  PDVAD - GET THE PDV ADDRESS
                                ;+
                                ; **-PDVAD-GET THE PDV ADDRESS
                                ;
                                ; THIS ROUTINE IS CALLED TO RETURN THE PDV INDEX AND ADDRESS
                                ; FOR A DDM.
                                ;
                                ; INPUTS:
                                ;       R5 - ADDRESS OF CONTEXT AREA WITN
                                ;       L$DDM(R5) = DDM NAME IN RAD50
                                ;
                                ; OUTPUTS:
                                ;       IF CC, PROCESS FOUND AND IS A DDM
                                ;          R2 = PDV INDEX
                                ;          R4 = PDV ADDRESS
                                ;       ELSE, ERROR
                                ;-

001334                          PDVAD:
001334  016502  000000                  MOV     L$DDM(R5),R2            ;; GET PROCESS NAME
001340                                  CALL    @PDVID                  ;; FIND IT'S PDV ID               ; RJK05
001344  103411                          BCS     10$                     ;; IF CC, LINE NOT IN SYSTEM
001346  010204                          MOV     R2,R4                   ;; COPY THE PDV INDEX
001350  067704  000000G                 ADD     @PDVTA,R4               ;; GET THE PDV ADDRESS            ; RJK05
001354  011404                          MOV     (R4),R4                 ;; POINT TO THE PDV
001356  132764  000001  000010          BITB    #ZF.DDM,Z.FLG(R4)       ;; IS THE PROCESS A DDM ?
001364  001001                          BNE     10$                     ;; IF NE, YES
001366  000261                          SEC                             ;; ELSE, ERROR
001370                          10$:    RETURN
```

```
                .TITLE  PRSEXE - PARSE EXECUTOR ACTION ROUTINES
                .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - PARSE COMMANDS ASSOCIATED WITH ''DEFINE EXECUTOR'' REQUESTS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                                                .SBTTL  .EXSEG - EXECUTOR SEGMENT BUFFER SIZE
                                        ;+
                                        ; **    .EXSEG - EXECUTOR SEGMENT BUFFER SIZE
                                        ;
                                        ; INPUTS:       TPARS CONVENTIONS.
                                        ;
                                        ; OUTPUT:       $ESEG   = PARSED EXECUTOR SEGMENT BUFFER SIZE
                                        ;-
001044                                  .EXSEG::
001044  016767  000000G 000000G                 MOV     .PNUMB,$ESEG    ; MOVE IT
001052  001403                                  BEQ     10$             ; IF EQ, ERROR
001054  005767  000000G                         TST     .PNUMH          ; TOO HIGH ?
001060  001405                                  BEQ     20$             ; IF EQ, OK

001062  012767  000000G 000000G 10$:            MOV     #ERRILN,$ERROR  ; error in number
001070  062716  000002                          ADD     #2,(SP)         ; else reject transition

001074  000207                          20$:    RETURN
```

```
      000000                                        .PSECT
                                            ;
                                            ; OFFSETS INTO SCRATCH BUFFER (L$SCR) FOR DMC BASE TABLE COUNTERS
                                            ;
      000124                                        .ASECT
              000000                                .=0
      000000                                B$5:    .BLKB   1                       ; BASE TABLE + 5
      000001                                B$6:    .BLKB   1                       ; BASE TABLE + 6
      000002                                B$7:    .BLKB   1                       ; BASE TABLE + 7
      000003                                B$10:   .BLKB   1                       ; BASE TABLE + 10
      000004                                B$11:   .BLKB   1                       ; BASE TABLE + 11
      000005                                B$12:   .BLKB   1                       ; BASE TABLE + 12
      000006                                B$7S10: .BLKB   1                       ; SUM OF BASE TABLE 7 AND 10
                                                    .EVEN
      000000                                        .PSECT
```

```
MD$FUL  000000          ME.CST  177765          ML$TOP  000001          MP$BRT  001620          MP$EVE  000311
MD$HAL  000001          ME.DIS  177755          MN$UNL  000377          MP$BSA  002176          MP$FNC  001752
MD$HEL  000002          ME.DON  177600          MO$ACC  000200          MP$BSD  002203          MP$GDT  002222
MD$KCP  000013          ME.FCO  177762          MO$ADD  000002          MP$BSI  002200          MP$GNM  002223
MD$KDP  000034          ME.FIO  177756          MO$ALA  000004          MP$BSP  006331          MP$GRO  002115
MD$KDZ  000036          ME.FOP  177763          MO$ALI  000000          MP$BUF  000170          MP$GRP  000541
MD$KL   000040          ME.FOR  177776          MO$CHA  000040          MP$BUP  006324          MP$GTY  002224
MD$KMX  000054          ME.FUN  177777          MO$CIR  000003          MP$BUS  001643          MP$HAD  001757
MD$KMY  000052          ME.GRO  177745          MO$CLE  000100          MP$CAC  001750          MP$HBT  002142
MD$LEN  000050          ME.HAR  177750          MO$COU  000060          MP$CAS  001762          MP$HDD  000162
MD$PCL  000011          ME.IID  177767          MO$DAC  000000          MP$CAT  002210          MP$HTM  001612
MD$QNA  000005          ME.LCO  177766          MO$DEF  000000          MP$CCS  004406          MP$HWA  002210
MD$UNA  000001          ME.LPR  177757          MO$DPR  000000          MP$CHN  002141          MP$IAT  001322
MD$WIT  000002          ME.MPR  177773          MO$ENT  000017          MP$CIR  000144          MP$IDE  000144
MD.ABO  000017          ME.MVE  177771          MO$EVE  000100          MP$CLK  002131          MP$IDP  006327
MD.ACC  000010          ME.OPE  177747          MO$INF  000160          MP$CLN  002126          MP$IHO  000215
MD.AOB  000016          ME.PLO  177751          MO$INS  000002          MP$CLT  002211          MP$INA  002177
MD.BOB  000011          ME.PMI  177743          MO$LIN  000001          MP$CMB  002142          MP$IND  002204
MD.DIA  000007          ME.PNA  177752          MO$LOG  000001          MP$CMK  000537          MP$INI  002201
MD.DOB  000015          ME.PRI  177775          MO$MIR  000031          MP$CMX  002153          MP$ITI  000776
MD.DUM  000             ME.PTY  177772          MO$NAM  000001          MP$CND  000310          MP$LAA  000231
MD.FAI  000014          ME.PVA  177760          MO$NIC  000023          MP$CNU  001753          MP$LAN  000234
MD.FNA  000001          ME.RES  177761          MO$NOD  000000          MP$COB  000311          MP$LAR  000202
MD.FOB  000006          ME.ROO  177754          MO$OFF  000001          MP$CON  002126          MP$LAS  000012
MD.LOA  000001          ME.SIZ  177774          MO$ON   000000          MP$COS  001604          MP$LBS  000203
MD.LSH  000020          ME.SYS  177746          MO$OPT  000001          MP$CPF  000230          MP$LCO  000226
MD.NOB  000012          ME.UCO  177770          MO$PER  000200          MP$CPL  000232          MP$LCT  000156
MD.NON  177777          MF$ACT  177776          MO$PRO  000002          MP$CPT  000226          MP$LHL  000232
MD.PER  000000          MF$ADD  000000          MO$PRS  000001          MP$CPU  000161          MP$LLE  000227
MD.RES  000004          MF$ADJ  177774          MO$REA  000200          MP$CSZ  001755          MP$LLO  004432
MD.ROB  000005          MF$ALL  177775          MO$SET  000000          MP$CUS  002127          MP$LMB  002152
MD.RSH  000013          MF$BYE  000300          MO$STA  000020          MP$CVA  000540          MP$LMX  002200
MD.SDU  000005          MF$CHA  000023          MO$SUM  000000          MP$DAL  002570          MP$LNA  000144
MD.SLO  000003          MF$DUM  000020          MO$TSK  000000          MP$DCO= 001464          MP$LNO  000233
MD.SNA  000000          MF$EVT  000001          MO$VOL  000000          MP$DDT  002177          MP$LOA  000170
MD.TLO  000004          MF$FUP  000264          MO$ZER  000000          MP$DEL  001131          MP$LOG  004514
MD.UNA  000002          MF$KNO  177777          MP$ACB  000012          MP$DES  000156          MP$LOO  000620
MD.UNR  000003          MF$LOA  000017          MP$ACC  000514          MP$DEV  002114          MP$LTM  001613
MD.UOB  000007          MF$LOO  177775          MP$ACT  004526          MP$DFA  001320          MP$LTY  004533
MD.VOL  000006          MF$REA  000024          MP$ADD  000766          MP$DHO= 001465          MP$LWI  000230
ME$ALI  000010          MF$SIG  177773          MP$ADJ  001440          MP$DIA  000173          MP$MAC  004420
ME$AL2  000003          MF$SPF  000302          MP$ADP  006325          MP$DLB  002571          MP$MAD  001630
ME$ARE  000005          MF$SYS  000026          MP$ADS  004406          MP$DLG  004521          MP$MAP  006323
ME$CIR  000003          MF$TES  000022          MP$ALB  000036          MP$DLI= 001466          MP$MAR  001635
ME$EXA  000000          MF$TRA  000301          MP$AMC  001640          MP$DLT  002200          MP$MAV  006321
ME$EXE  000200          MF$TRI  000021          MP$AMH  001641          MP$DRO  001441          MP$MBN  001636
ME$LIN  000001          MF$ZER  000025          MP$ANB  004420          MP$DST  000454          MP$MBR  001637
ME$LOG  000002          ML$ALL  100000          MP$ASB  000024          MP$DTE  002140          MP$MBU  001642
ME$MOD  000004          ML$CLS  000000          MP$ASC  004432          MP$DTY= 001452          MP$MCB  000156
ME$NOD  000000          ML$CON  000001          MP$AUS  000512          MP$DUA  000207          MP$MCO  001632
ME$NON  177777          ML$EXT  000001          MP$BBT  002165          MP$DUC  000210          MP$MDE  004421
ME$OBJ  000007          ML$FIL  000002          MP$BDF  002164          MP$DUM  000202          MP$MHO  001633
ME$OB2  000004          ML$FIR  000000          MP$BFQ  002121          MP$DUP  002127          MP$MLB  000202
ME$PRO  000005          ML$INT  000000          MP$BLK  001616          MP$DVC  002114          MP$MLK  001306
ME$SYS  000006          ML$KNO  140000          MP$BLO  001452          MP$DWE  001321          MP$MLN  001631
ME.BLO  177744          ML$MON  000003          MP$BMX  002176          MP$ELT  000157          MP$MLP  006333
ME.CON  177753          ML$SYS  000002          MP$BNP  006330          MP$ETY  001605          MP$MRB  002171
```

```
                                                .SBTTL  CKEVEB - CHECK EVENT RANGE BEGINNING
                                        ;+
                                        ;
                                        ; ***  CKEVEB - CHECK EVENT RANGE BEGINNING
                                        ;
                                        ; FUNCTION:
                                        ;
                                        ;   ENSURE THAT NUMBER IS GROSSLY WITHIN RANGE AND SAVE IT AS IF IT WERE
                                        ;   THE BEGINNING OF A RANGE OF NUMBERS.
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;   .PNUMB - LOW ORDER WORD OF PARSED NUMBER
                                        ;   .PNUMH - HIGH ORDER WORD OF PARSED NUMBER
                                        ;
                                        ; OUTPUTS:
                                        ;
                                        ;   $EVBEG- PARSED NUMBER
                                        ;
                                        ; REGISTERS R3, R4, R5 ARE PRESERVED
                                        ;-
                                        ;
                                        ; PROCEDURE BODY
                                        ;
 000064                                 CKEVEB::
 000064  016700  000000G                        MOV     .PNUMB,R0               ;GET NUMBER
 000070  002406                                 BLT     10$                     ;BR IF NUMBER < 0
 000072  005767  000000G                        TST     .PNUMH                  ;CHECK HIGH ORDER WORD
 000076  001003                                 BNE     10$                     ;BR IF NOT WORD VALUE
 000100  010067  000000G                        MOV     R0,$EVBEG               ;SAVE 1ST NUMBER TO WAIT FOR 2ND
 000104  000402                                 BR      20$                     ;FINISH UP
 000106  062716  000002                 10$:    ADD     #2,(SP)                 ;REJECT THE TRANSITION
 000112  000207                         20$:    RETURN
```

```
                                                    .SBTTL  .MOCLT - PARSE CLEAR TIMER
                                            ;+
                                            ;** .MOCLT - PARSE CLEAR TIMER
                                            ;
                                            ; INPUTS        .TPARS VARIABLES.
                                            ;
                                            ; OUTPUT        $CLT = PARSED CLEAR TIMER
                                            ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                            ;-
000104                                      .MOCLT::
000104  005767  000000G                             TST     .PNUMH          ; HIGH ORDER WORD
000110  001007                                      BNE     10$             ; REJECT IF TOO HIGH
000112  105767  000001G                             TSTB    .PNUMB+1        ; TEST HIGH BYTE
000116  001004                                      BNE     10$             ; BR IF  ILLEGAL NUMBER
000120  116767  000000G 000000G                     MOVB    .PNUMB,$CLT     ; MOVE VALUE TO PARSED AREA
000126  000207                                      RETURN
000130  000167  000352                      10$:    JMP     ILLNUM
```

```
                                        .SBTTL  .MOCNN  - PARSE CHANNEL NUMBER
                                ;+
                                ; ** .MOCNN - PARSE CHANNEL NUMBER
                                ;
                                ; INPUTS        .TPARS VARIABLES.
                                ;
                                ; OUTPUT        THE PARSED NUMBER IS PUT INTO THE CHANNEL LIST.
                                ;               THE TRANSITION IS REJECTED IF THERE IS NO ROOM.
                                ;
                                ;-

000560                          .MOCNN::
000560  005767  000000G                 TST     .PNUMH                  ; TEST HIGH ORDER WORD
000564  001350                          BNE     ILLNUM                  ; IF NE, TOO HIGH
000566  016701  000000G                 MOV     .PNUMB,R1               ; GET NUMBER
000572  001745                          BEQ     ILLNUM                  ; IF EQ, TOO LOW
000574  020127  007777                  CMP     R1,#4095.               ; CHECK HIGH BOUND
000600  101342                          BHI     ILLNUM                  ; IF HI, REJECT TRANSITION
000602  012700  000000G                 MOV     #$CHNLS,R0              ; POINT TO CHANNEL LIST
000606  062710  000002                  ADD     #2,(R0)                 ; ALLOCATE AN ENTRY
000612  061000                          ADD     (R0),R0                 ; POINT TO FIRST FREE SPACE
000614  020027  000000G                 CMP     R0,#$CHNND              ; IS THERE ROOM ?
000620  101002                          BHI     CHNERR                  ; IF HI, NO
000622  010110                          MOV     R1,(R0)                 ; LOAD UP
000624  000207                          RETURN

000626  012767  000000G 000000G CHNERR: MOV     #ERRCHN,$ERROR          ; ERROR IN CHANNEL LIST
000634  000727                          BR      REJECT

```

```
                                        ;****
                                        ; INITIALIZE TABLE GENERATION
                                        ;****

        000170                                  ISTAT$  UPDSTB,UPDKTB
                                        ;
                                        ; SEARCH TABLE FOR DEFINITION
                                        ;
        000170                                  STATE$
        000170                                   TRAN$  %BUF$DF%,$EXIT,$UPBUF
        000170                                   TRAN$  %FEA$DF%,$EXIT,$UPFEA
        000170                                   TRAN$  %DEC$DF%,$EXIT,$UPDEC
        000170                                   TRAN$  %LLC$DF%,$EXIT,$UPLLC
        000170                                   TRAN$  %DLC$DF%,$EXIT,$UPDLC
        000170                                   TRAN$  %DDM$DF%,$EXIT,$UPDDM
        000170                                   TRAN$  %CNT$DF%,$EXIT,$UPCNT
        000170                                   TRAN$  %SLT$DF%,$EXIT,$UPSLT
        000170                                   TRAN$  %UNT$DF%,$EXIT,$UPUNT
        000170                                   TRAN$  %NOD$DF%,$EXIT,$UPNOD
        000170                                   TRAN$  %PAR$DF%,$EXIT,$UPPAR
        000170                                   TRAN$  %STA$DF%,$EXIT,$UPSTA
        000170                                   TRAN$  %OBJ$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %LOG$ST%,$EXIT,$UPLOG
        000170                                   TRAN$  %EVT$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %REM$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %SER$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %FIL$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %ROU$DF%,$EXIT,$UPROU
        000170                                   TRAN$  %PSN$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %X3P$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %DTE$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %CHN$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %PVC$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %CUG$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %DST$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %DSA$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %DSC$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %RDT$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %X2P$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %SVC$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  %X29$DF%,$EXIT,SKIPIT
        000170                                   TRAN$  !END,$EXIT,CLENUP
        000170                                   TRAN$  $LAMDA,$EXIT,$MOVIT
                                        ; "END"
        000170                                  STATE$  END
        000170                                   TRAN$  "END",DOLLAR
                                        ; "$"
        000170                                  STATE$  DOLLAR
        000170                                   TRAN$  '$,DF
                                        ; "DF"
        000170                                  STATE$  DF
        000170                                   TRAN$  "DF",$EXIT
                                        ;
                                        ; FINAL STATE
                                        ;
        000170                                  STATE$
```

```
                                                                              N  2
CFEUPD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:15      PAGE 3
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME        REFERENCES

CLOSE$            #4-89
DBGTP$            #5-138     #5-144     #5-145     #5-146     #5-147     #5-148     #5-149     #5-150     #5-151     #5-152
                  #5-153     #5-154     #5-155     #5-156     #5-157     #5-158     #5-159     #5-160     #5-161     #5-162
                  #5-163     #5-164     #5-165     #5-166     #5-167     #5-168     #5-169     #5-170     #5-171     #5-172
                  #5-173     #5-174     #5-175     #5-176
DIR$              #4-90      #13-397     13-397     13-400    #13-409     13-409
EPRINT            #4-75       6-243
ERROR$            #4-64       6-214      6-230      6-250
GTIM$S            #4-90       13-409
GTSK$S            #4-90       13-397
ISTAT$            #4-89       5-139
MOV$              #13-397     13-397    #13-409     13-409
MTRAN$            #5-138
OFF$              #13-397    #13-409
OPEN$R            #4-89
PRINT             #4-59
STATE$            #4-89       5-142     #5-178     #5-181     #5-184     #5-189
TRAN$             #4-89      #5-143     #5-144     #5-145     #5-146     #5-147     #5-148     #5-149     #5-150     #5-151
                  #5-152     #5-153     #5-154     #5-155     #5-156     #5-157     #5-158     #5-159     #5-160     #5-161
                  #5-162     #5-163     #5-164     #5-165     #5-166     #5-167     #5-168     #5-169     #5-170     #5-171
                  #5-172     #5-173     #5-174     #5-175     #5-176     #5-179     #5-182     #5-185
```

```
                                        .TITLE  EVTTBL - EVENT LOGGING TABLES
                                        .IDENT  /V05.00/
                                        .ENABL  LC
          000012                        .RADIX  10

                                ;
                                ; Copyright (c) 1982, 1983, 1985 by
                                ; Digital Equipment Corporation, Maynard, Mass.
                                ;
                                ;
                                ; This software is furnished under a license and may be used and  copied
                                ; only  in  accordance  with  the  terms  of  such  license and with the
                                ; inclusion of the above copyright notice.  This software or  any  other
                                ; copies  thereof may not be provided or otherwise made available to any
                                ; other person.  No title to and ownership of  the  software  is  hereby
                                ; transferred.
                                ;
                                ; The information in this software is subject to change  without  notice
                                ; and  should  not  be  construed  as  a commitment by Digital Equipment
                                ; Corporation.
                                ;
                                ; Digital assumes no responsibility for the use or  reliability  of  its
                                ; software on equipment which is not supplied by Digital.
                                ;
                                ;
                                ; Module Description:
                                ;
                                ; CFE/NCP/VNP - Event Parameter Table
                                ;
                                ;
                                ; Distributed Systems Software Engineering
                                ;
                                ; Ident history:
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;



                                        .SBTTL  Macro calls

          000001                        ...N            = 1
                                        .IRP    ...,<0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15>
                                        $$$B'...        = ...N
                                        ...N = 2*...N
                                        .ENDR

                                        .MACRO  ...BIS  W,B
                                        $$$W'W = $$$W'W ! $$$'B
```

```
PRMPTS - PROMPTS, STRING ASSEMB MACRO V05.03b Saturday 29-Jun-85 17:01
Table of contents
```

```
     4-    59     MACROS AND SYMBOLS AND INITIALIZATION
     5-    73     PROMPTS
```

```
	.IF NDF	$GOFF
	.TITLE	PRMPTS - PROMPTS, STRING ASSEMBLY
	.IFF
	.TITLE	PRMPTO - PROMPTS, OFFSET ASSEMBLY
	.ENDC

	.IDENT	/V05.00/
	.ENABL	LC
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1984, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;	NETWORK MANAGEMENT - PROMPTS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 2.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;
;

```

```
     56                                          ;***
     57                                          ; LOCAL DATA
     58                                          ;***
     59
     60                                                  .ENABL  LC
     61                                                  .NLIST  BEX
     62           000002                                 EX$ERR=2
     63 000000       002      103       106      ERRNAM: .ASCIZ  <EX$ERR>/CFE -- Nooe name already in use/
     64 000041       002      103       106      ERRADD: .ASCIZ  <EX$ERR>/CFE -- Node address already in use/
     65 000105       002      103       106      ERREXE: .ASCIZ  <EX$ERR>/CFE -- Executor node not allowed/
     66                                                  .LIST BEX
     67                                                  .EVEN
     68                                          ;
     69                                          ; DEFINE REMOTE BLOCK OFFSETS
     70                                          ;
     71 000000                                           .ASECT
     72           000000                                 .=0
     73 000000                                           .BLKW   1               ; C.LNK
     74 000002                                           .BLKW   1               ; C.STS
     75 000004                                   C.NAM:  .BLKW   2               ; REMOTE NODE NAME
     76 000010                                   C.ADD:  .BLKW   1               ; REMOTE NODE ADDRESS
     77
     78 000150                                           .PSECT
```

```
                                        .SBTTL  FNDNAM - FIND NODE NAME
                                ;+
                                ; *** - FNDNAM - FIND NODE NAME
                                ;
                                ; INPUTS:
                                        $NODNA = 2 WORD RAD50 NODE NAME
                                ;
                                ; OUTPUTS:
                                        C-BIT = SUCCESS/FAILURE
                                        R0 = TEMPLATE ADDRESS IF SUCCESS
                                ;
                                ;-
001254                          FNDNAM:
001254  012700  000000G                 MOV     #$TMLST,R0      ; GET START OF TEMPLATES
001260  011000                  10$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
001262  000261                          SEC                     ; ASSUME FAILURE
001264  001415                          BEQ     20$             ; IF EQ, NO MORE .. ERROR
001266  122760  000000G 000000G         CMPB    #CS.REM,C.STS(R0) ; IS THIS A REMOTE TEMPLATE ?
001274  001371                          BNE     10$             ; IF NE, NO
001276  026760  000000G 000004          CMP     $NODNA,C.NAM(R0) ; IS THIS THE RIGHT TEMPLATE ?
001304  001365                          BNE     10$             ; IF NE, NO .. KEEP LOOKING
001306  026760  000002G 000006          CMP     $NODNA+2,C.NAM+2(R0) ; ARE WE SURE ?
001314  001361                          BNE     10$             ; IF NE, NO
001316  000241                          CLC                     ; INDICATE SUCCESS
001320  000207                  20$:    RETURN

```

```
                                         .SBTTL  PRSCIR - PARSE AN ASCII CIRCUIT-ID
                                   .IF DF R$$11S
                                         .SBTTL  PRSLIN - PARSE AN ASCII LINE-ID
                                   .IFTF
                                  ;+
                                  ; **-PRSCIR-PARSE AN ASCII CIRCUIT-ID
                                   .IFT
                                  ; **-PRSLIN-PARSE AN ASCII LINE-ID
                                   .ENDC
                                  ;
                                  ; THIS ROUTINE PARSES AN ASCII IMAGE CIRCUIT-ID STRING INTO
                                  ; DEVICE, CONTROLLER, UNIT AND TRIBUTARY NUMBERS. WILD CARDS
                                  ; ARE ALLOWED IN CONTROLLER, UNIT AND TRIBUTARY FIELDS.
                                  ;
                                  ; INPUTS:
                                  ;       R1 = ADDRESS OF IMAGE CIRCUIT-ID STRING
                                  ;       R4 = ADDRESS OF OUTPUT PARAMETER BLOCK
                                  ;
                                  ; OUTPUTS:
                                  ;       C-BIT CLEAR:
                                  ;         R1 = ADDRESS OF NEXT BYTE AFTER CIRCUIT-ID (unless X.25 circuit)
                                  ;         R4 = ADDRESS OF THE PARAMETER BLOCK
                                  ;
                                  ;       C-BIT SET:
                                  ;         INVALID CIRCUIT-ID SYNTAX OR UNRECOGNIZED CIRCUIT (ERROR CODE IN R0)
                                  ;
                                  ;       For X.25 permanent circuits, L$SLT contains the unmapped address
                                  ;       of the PVC name block.
                                  ;       For new X.25 permanent circuits, L$SLT is 0 and L$TPT is -1.
                                  ;
                                  ; REGISTERS:
                                  ;       R3, R4, R5 ARE PRESERVED
                                  ;-

000004                            PRSCIR::
                                   .IF DF R$$11S
                                          TST     (PC)+                    ; CLEAR CARRY FOR CIRCUIT

                                  PRSLIN::
                                          SEC                              ; SET CARRY FOR LINE
                                          ROR     ENTTYP                   ; SAVE ENTITY INDICATOR
                                   .ENDC
000004                                    CALL    $SAVAL                   ; SAVE ALL REGISTERS
000010  010405                            MOV     R4,R5                    ; COPY BUFFER POINTER
000012  012700  000026                    MOV     #<L$PDV/2>,R0            ; SET LENGTH OF BUFFER TO CLEAR
000016  005024                    10$:    CLR     (R4)+                    ; INITIALIZE OUTPUT AREA
000020                                    SOB     R0,10$                   ; ...

                                          .IF DF  S$$BAS ! R$$RTR

                                          CALL    $TRCIR                   ; Translate logical name
                                          BCC     120$                     ; if CC, parsing all done !

                                          .ENDC   ; DF S$$BAS ! R$$RTR

                                          .IF NDF R$$11S & R$$PRO

```

```
                                        .SBTTL  CHKWLD - CHECK FOR WILD CARD FIELD
                                ;+
                                ; **-CHKWLD-CHECK FOR WILD CARD FIELD
                                ;
                                ; THIS ROUTINE CHECKS FOR LEGAL WILD CARD SYNTAX.
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS OF NEXT BYTE IN STRING
                                ;       R3 = LEGAL TERMINATING CHARACTER
                                ;       R4 = ADDRESS OF THE END OF THE STRING + 1
                                ;
                                ; OUTPUTS:
                                ;       C-BIT CLEAR:
                                ;       R0 = ADDRESS OF NEXT BYTE IN STRING
                                ;       R2 = COMPLETION CODE ( 0 = MORE IN THE STRING)
                                ;                            ( NOT 0 = NO MORE IN THE STRING)
                                ;       C-BIT SET:
                                ;       R2 = COMPLETION CODE ( 0 = * NOT FOUND)
                                ;                            ( 1 = * FOUND, NOT FOLLOWED BY VALID TERMINATOR)
                                ;
                                ; REGISTERS:
                                ;       R3, R4, R5 ARE PRESERVED
                                ;-

001372                          CHKWLD:
001372  005002                          CLR     R2                      ; ASSUME '*' NOT FOUND
001374  121027  000052                  CMPB    (R0),#'*                ; WILD CARD ?
001400  001013                          BNE     10$                     ; IF NE, NO - RETURN
001402  005200                          INC     R0                      ; SKIP OVER CHARACTER
001404  005202                          INC     R2                      ; ASSUME END OF STRING
001406  020400                          CMP     R4,R0                   ; AT END OF STRING YET ?
001410  001410                          BEQ     20$                     ; IF EQ, YES
001412  122003                          CMPB    (R0)+,R3                ; LEGAL TERMINATOR ?
001414  001005                          BNE     10$                     ; IF NE, NO - RETURN
001416  020400                          CMP     R4,R0                   ; AT END OF STRING ?
001420  001404                          BEQ     20$                     ; IF EQ, YES - RETURN
001422  103402                          BLO     10$                     ; IF LO, ERROR
001424  005002                          CLR     R2                      ; ELSE, INDICATE MORE IN STRING
001426  000401                          BR      20$                     ; AND RETURN

001430  000261                  10$:    SEC
001432                          20$:    RETURN

                                        .DSABL  LSB

        000001                          .END
```

```
                         ;****
                         ; LOCAL MACROS
                         ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT

                         ;****
                         ; LOCAL DATA
                         ;****

        000040           STRMX   = 32.                  ; MAXIMUM OF 32. CHARS IN A NODE ID STRING
        000004           C.ADD   = 4                    ; OFFSET OF MAX NODE ADDRESS IN ROUTING TEMPLATE
```

```
                                                   .SBTTL  .EXINA - EXECUTOR INACTIVITY TIMER
                                          ;+
                                          ; **     .EXINA - EXECUTOR INACTIVITY TIMER
                                          ;
                                          ; INPUTS:        TPARS CONVENTIONS.
                                          ;
                                          ; OUTPUT:        $EINA    = PARSED EXECUTOR INACTIVITY TIMER
                                          ;-
001076                                    .EXINA::
001076  116767  000000G 000000G                    MOVB    .PNUMB,$EINA      ; MOVE IT
001104  001406                                     BEQ     10$               ; IF EQ, ERROR
001106  005767  000000G                            TST     .PNUMH            ; TOO HIGH ?
001112  001003                                     BNE     10$               ; IF EQ, OK
001114  105767  000001G                            TSTB    .PNUMB+1          ; BYTE VALUE?
001120  001405                                     BEQ     20$               ; BR IF YES - OK

001122  012767  000000G 000000G   10$:             MOV     #ERRILN,$ERROR    ; ERROR IN NUMBER
001130  062716  000002                             ADD     #2,(SP)           ; ELSE REJECT TRANSITION

001134  000207                    20$:             RETURN
```

```
                                        .SBTTL  BIT DEFINITIONS

                                ;
                                ; PARSE FLAG DEFINITIONS (L$PFG)
                                ;
        000001                          LP$UNT = 1                      ; UNIT NUMBER FOUND
        000002                          LP$TRB = 2                      ; TRIBUTARY NUMBER FOUND
        000004                          LP$MUX = 4                      ; DEVICE IS MUX
        000010                          LP$MPT = 10                     ; LINE IS MULTIPOINT
        000020                          LP$WDV = 20                     ; WILD CARD DEVICE NAME FOUND
        000040                          LP$WCN = 40                     ; WILD CARD CONTROLLER NUMBER FOUND
        000100                          LP$WUN = 100                    ; WILD CARD UNIT NUMBER FOUND
        000200                          LP$WTR = 200                    ; WILD CARD TRIBUTARY NUMBER FOUND
        000360                          LP$WLD = LP$WDV!LP$WCN!LP$WUN!LP$WTR ; WILD CARD FIELD MASK
                                ;
                                ; FLAGS WORD BIT DEFINITIONS (L$FLG)
                                ;
        000001                          LF$REA = 1                      ; READ COUNTERS OPERATION
        000002                          LF$ZER = 2                      ; ZERO COUNTERS OPERATION
        000004                          LF$SKP = 4                      ; SKIP NEXT "FIND NEXT LINE" OPERATION.
                                                                        ; THIS IS USED TO FORCE AN EXTRA PASS
                                                                        ; FOR A MULTIPOINT LINE TO RETURN THE
                                                                        ; CONTROLLER COUNTERS AS WELL AS ALL
                                                                        ; OF THE TRIBUTARY COUNTERS.
        000010                          LF$VR2 = 10                     ; CONNECTED TO VERSION 2.0 NCP
        000020                          LF$END = 20                     ; EXECUTOR IS AN ENDNODE
        000040                          LF$SIG = 40                     ; SHOWING SIGNIFIGANT LINES/CIRCUITS
        040000                          LF$MLT = 40000                  ; MULTIPLE ADJACENCY FLAG
        100000                          LF$SEG  = 100000                ; SEGMENTED RESPONSE IN PROGRESS
                                ;
                                ; BIT POSITION DEFINITIONS
                                ;
        000001                          BIT0 = 1
        000002                          BIT1 = 2
        000004                          BIT2 = 4
        000010                          BIT3 = 10
        000020                          BIT4 = 20
        000040                          BIT5 = 40
        000100                          BIT6 = 100
        000200                          BIT7 = 200
        000400                          BIT8 = 400
        001000                          BIT9 = 1000
        002000                          BIT10 = 2000
        004000                          BIT11 = 4000
        010000                          BIT12 = 10000
        020000                          BIT13 = 20000
        040000                          BIT14 = 40000
        100000                          BIT15 = 100000
   64
```

```
                                                          N 13
PRSLIN - PARSE A LINE-ID        MACRO V05.03b Saturday 29-Jun-85 13:23  Page 13-3
Symbol table

MP$MRO  001605       MP$RET  001631       MP$USR  001750       MT$CON  000001       M$$MGE= 000000
MP$MRP  006322       MP$RFA  001323       MP$VEC  004410       MT$DMC  000004       M$$NET= 000000
MP$MRT  002153       MP$RMX  002201       MP$VFR  004406       MT$ETH  000006       M$$OVR= 000000
MP$MRV  006320       MP$RPA  004374       MP$WDF  002165       MT$HEX  000040       NM.CLN  003400
MP$MSB  000170       MP$RPR  001606       MP$WMX  002177       MT$LPB  000005       NM.INI  002400
MP$MVE  000145       MP$RRT  002213       MP$XMT  002166       MT$MAX  000037       NM.OPR  003000
MP$MVI  001634       MP$RST  002212       MP$XPF  000242       MT$MUL  000100       NM.VR2  002401
MP$MVR  001751       MP$RSV  001754       MP$XPL  000244       MT$NLE  000017       NM.VR3  002402
MP$MWN  002154       MP$RSZ  001756       MP$XPT  000240       MT$NON  000001       N$$ACC= 000001
MP$MXB  002172       MP$RTI  001616       MP$XXX  177777       MT$NR4  000005       N$$BUF= 000001
MP$MXC  000466       MP$RTM  001611       MP$5ST  004406       MT$NTY  000060       N$$LDV= 000001
MP$MXR  001630       MP$RVE  001604       MS$ACT  000001       MT$OCT  000060       N$$MCP= 000001
MP$MXW  002143       MP$RVT  000156       MS$ADU  000010       MT$PHA  000002       N$$MLL= 000001
MP$NAA  006335       MP$SAC  000514       MS$ALO  000007       MT$POI  000000       N$$MOV= 000010
MP$NAC  001130       MP$SAD  000543       MS$ASE  000006       MT$QP2  000010       N$$NCT= 000001
MP$NAP  006326       MP$SCA  000310       MS$ATR  000011       MT$ROU  000000       N$$PEM= 000001
MP$NCT  000240       MP$SCO  000144       MS$AUT  000000       MT$R04  000004       PDVAD   000662R
MP$NET  002114       MP$SCT  002176       MS$CLE  000003       MT$SEC  000000       PDVID = ******  GX
MP$NLI  000765       MP$SDU  000203       MS$DED  000004       MT$SGD  000020       PDVTA = ******  GX
MP$NNA  000764       MP$SDV  000160       MS$DIE  000003       MT$SYS  000002       PRSLIN  000000RG
MP$NOD  000500       MP$SEH  004401       MS$DUM  000004       MT$TER  000001       P$$P45= 000000
MP$NRA  006334       MP$SER  000144       MS$FAI  000013       MT$TRI  000002       P$$WRD= 000000
MP$NRB  000214       MP$SET  000000       MS$GLO  000000       MT$TYP  007777       Q$$OPT= 000010
MP$NSA  001617       MP$SGZ  001644       MS$HOL  000002       MT$USD  000000       R$R0  = ******  GX
MP$NTI  002141       MP$SID  000176       MS$INA  000002       MT$X25  000003       R$R1  = ******  GX
MP$NUM  001642       MP$SIN  000310       MS$LOA  000003       MU$INC  000001       R$$DER= 000000
MP$NVE  001274       MP$SLI  000156       MS$LOO  000002       MU$OUT  000002       R$$K11= 000001
MP$NXN= 001476       MP$SLO  000171       MS$OFF  000001       MU$PER  000000       R$$SND= 000000
MP$OAC  000632       MP$SMX  002202       MS$ON   000000       MV$III  000000       R$$11M= 000000
MP$OAN  000620       MP$SND  000500       MS$PRO  000006       MV$IV   000001       SF$ANM  000001
MP$OCO  000776       MP$SNP  006332       MS$REA  000004       MX$ACT  000020       SF$DNM  000002
MP$OHO  000214       MP$SNU  000542       MS$REF  000001       MX$CIR  000020       SF.ACT= 000200
MP$ONA  000764       MP$SOB  000524       MS$RES  000003       MX$CLN  000020       SF.ENA= 000100
MP$ONR  002114       MP$SPA  000157       MS$RST  000001       MX$CNM  000006       SF.LPB= 000004
MP$OQL  001453       MP$SPR  000536       MS$RSX  000002       MX$CON  000006       SF.MFL= 000040
MP$OTI  000777       MP$SPS  000513       MS$RT   000005       MX$DAC  000020       SF.PAC= 000020
MP$OTY  001022       MP$STA  000000       MS$SER  000002       MX$DTE  000020       SF.REA= 000010
MP$OUS  000777       MP$STI  002140       MS$SHU  000002       MX$FIL  000034       SF.SER= 000001
MP$OVE  001010       MP$STT  002201       MS$SRV  000007       MX$LGC  000777       SF.SVC= 000002
MP$OWN  004374       MP$STY  000175       MS$STA  000000       MX$LIN  000020       SF.UNL= 000040
MP$PAR  000036       MP$SUB  000001       MS$SYN  000012       MX$LON  000006       SLTMA = ******  GX
MP$PAS  000513       MP$SUR  000156       MS$TER  000001       MX$NMS  000454       SLTNM = ******  GX
MP$PCH  002152       MP$SUS  000512       MS$TOP  000003       MX$NOD  000006       S$$WRG= 000000
MP$PCO  000024       MP$SVR  000163       MS$TRI  000005       MX$OBJ  000006       S$$YSZ= 007600
MP$PCT  000144       MP$SWI  002260       MS$UNR  000005       MX$OWN  000040       S.COST  000001
MP$PDT  002114       MP$SYP  002424       MS$VMS  000004       MX$PAR  000036       S.FLG   000000
MP$PHA  000170       MP$TFL  000156       MS.DON  177600       MX$PAS  000010       S.LEN   000004
MP$PHY  000012       MP$TH1  002206       MS.MOR  000002       MX$RAC  000047       S.NMST  000002
MP$PLI  000025       MP$TH2  002205       MS.PAR  000003       MX$RID  000047       S.OWNR  000003
MP$PLN  002140       MP$TH3  002202       MS.SUC  000001       MX$RPS  000047       T$$KMG= 000000
MP$PLO  000012       MP$TLN  000202       MT$AR4  000003       MX$SID  000040       T$$MIN= 000000
MP$PMC  004374       MP$TLO  000172       MT$ASC  000100       MX$SNK  000377       V$$CTR= 001000
MP$PNT  002126       MP$TPA  004375       MT$BIL  000001       MX$TYP  000077       X$$DBT= 000000
MP$PRI  004411       MP$TRI  002164       MT$BYS  000011       MX$UID  000020       ZF.COU= 001000
MP$PRO  002130       MP$TST  000144       MT$CI   000007       M$$CRB= 000124       ZF.DOM= 000001
MP$PSS  001763       MP$TYP  002130       MT$CLE  000077       M$$CRX= 000000       ZF.DIA= 004000
MP$PST  001762       MP$UCS  004407       MT$COD  000200       M$$FCS= 000000       ZF.DLC= 000002
```

```
                                                 .SBTTL  CKEVEC - CHECK EVENT CLASS
                                         ;+
                                         ;
                                         ; ***  CKEVEC - CHECK EVENT CLASS
                                         ;
                                         ; FUNCTION:
                                         ;
                                         ;   CHECK RANGE OF EVENT CLASS, SAVE IT, AND INITIALIZE EVENT MASK.
                                         ;
                                         ; INPUTS:
                                         ;
                                         ;   .PNUMB - LOW ORDER WORD OF PARSED NUMBER
                                         ;   .PNUMH - HIGH ORDER WORD OF PARSED NUMBER
                                         ;
                                         ; OUTPUTS:
                                         ;
                                         ;   $EVENT - 32 ZERO BITS FOR EVENT MASK
                                         ;   $CLASS - EVENT CLASS (BITS 6-15)
                                         ;
                                         ; REGISTERS R3, R4, R5 ARE PRESERVED
                                         ;-
                                         ;
                                         ; PROCEDURE BODY
                                         ;
000114                                   CKEVEC::
000114  005767  000000G                          TST     .PNUMH                  ;CHECK HIGH ORDER WORD
000120  001026                                   BNE     10$                     ;BR IF HIGH ORDER WORD <> 0
000122  016700  000000G                          MOV     .PNUMB,R0               ;GET NUMBER
000126  002423                                   BLT     10$                     ;BR IF CLASS < 0
000130  020027  000777                           CMP     R0,#511.                ;COMPARE CLASS TO 511.
000134  003020                                   BGT     10$                     ;BR IF CLASS <= 511.
        000006                                   .REPT   6
                                                 ASL     R0                      ;GET EVENT CLASS IN BITS 6-15
                                                 .ENDR
000152  010067  000000G                          MOV     R0,$CLASS               ;SAVE CLASS FOR LATER
000156  005000                                   CLR     R0                      ;SET UP TO ZERO BIT ACCUMULATOR
000160  012701  000000G                          MOV     #$EVENT,R1              ;ADDRESS OF BIT ACCUMULATOR
000164  012702  000010                           MOV     #8.,R2                  ;ZERO 64 BITS WORTH
000170  004767  000074                           CALL    SPRAY                   ;DO IT
000174  000402                                   BR      20$
000176  062716  000002                   10$:    ADD     #2,(SP)                 ;REJECT TRANSITION
000202  000207                           20$:    RETURN
```

```
                                                .SBTTL  .MODFD - PARSE DEFAULT DATA
                                        ;+
                                        ; ** .MODFD - PARSE DEFAULT DATA
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $DFD = PARSED DEFAULT DATA
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;-
000134                                  .MODFD::
000134  005767  000000G                         TST     .PNUMH          ; HIGH ORDER WORD
000140  001162                                  BNE     ILLNUM          ; REJECT IF TOO HIGH
000142  016700  000000G                         MOV     .PNUMB,R0       ; GET PARSED VALUE
000146  020027  000025                          CMP     R0,#21.         ; CHECK LOW BOUND
000152  103555                                  BLO     ILLNUM          ; IF LO, OUT OF RANGE
000154  020027  002005                          CMP     R0,#1029.       ; CHECK HIGH BOUND
000160  101152                                  BHI     ILLNUM          ; IF HI, OUT OF RANGE
000162  010067  000000G                         MOV     R0,$DFD         ; MOVE VALUE TO PARSED AREA
000166  000207                                  RETURN
```

```
B  1              J  5              E 10              M 14
C  1              K  5              F 10              N 14
D  1              L  5  PRMPTO      G 10              B 15
E  1              M  5              H 10              C 15
F  1              N  5              I 10              D 15
G  1              B  6              J 10              E 15
H  1              C  6              K 10  PRSEXE      F 15
I  1  CFEUPD      D  6              L 10              G 15
J  1              E  6              M 10              H 15  PRSMOD
K  1              F  6              N 10              I 15
L  1              G  6              B 11              J 15
M  1              H  6              C 11              K 15
N  1              I  6              D 11              L 15
B  2              J  6              E 11              M 15
C  2              K  6  PRSACT      F 11              N 15
D  2              L  6              G 11              B 16
E  2              M  6              H 11              C 16
F  2              N  6              I 11              D 16
G  2              B  7              J 11              E 16
H  2              C  7              K 11              F 16
I  2              D  7              L 11              G 16
J  2              E  7              M 11              H 16
K  2              F  7              N 11              I 16
L  2              G  7              B 12              J 16
M  2              H  7              C 12              K 16
N  2              I  7              D 12              L 16
B  3  CFEUSB      J  7              E 12              M 16
C  3              K  7              F 12
D  3              L  7              G 12
E  3              M  7              H 12
F  3              N  7              I 12  PRSLIN
G  3              B  8              J 12
H  3              C  8              K 12
I  3              D  8              L 12
J  3              E  8              M 12
K  3              F  8              N 12
L  3  EVTTBL      G  8  PRSCIR      B 13
M  3              H  8              C 13
N  3              I  8              D 13
B  4              J  8              E 13
C  4              K  8              F 13
D  4              L  8              G 13
E  4              M  8              H 13
F  4              N  8              I 13
G  4              B  9              J 13
H  4  NOSNA       C  9              K 13
I  4              D  9              L 13
J  4              E  9              M 13
K  4              F  9              N 13
L  4              G  9              B 14
M  4  PRMPTS      H  9              C 14
N  4              I  9              D 14
B  5              J  9              E 14
C  5              K  9              F 14
D  5              L  9              G 14
E  5              M  9              H 14  PRSLOG
F  5              N  9              I 14
G  5              B 10              J 14
H  5              C 10              K 14
I  5              D 10              L 14
```

```
                                          .SBTTL  .MOCRN  - MARK CHANNEL RANGE
                                  ;+
                                  ; ** .MOCRN - MARK CHANNEL RANGE
                                  ;
                                  ; INPUTS        $CHNLS = CHANNEL LIST
                                  ;
                                  ; OUTPUT        THE LAST CHANNEL IN THE LIST IS MARKED AS
                                  ;               THE BEGINNING OF A RANGE. THE TRANSITION IS REJECTED
                                  ;               IF ANOTHER RANGE IS CURRENTLY ACTIVE.
                                  ;
                                  ;-
000636                            .MOCRN::
000636  012700  000000G                   MOV     #$CHNLS,R0      ; POINT TO CHANNEL LIST
000642  061000                            ADD     (R0),R0         ; POINT TO CURRENT ENTRY
000644  005760  177776                    TST     -2(R0)          ; CHECK FOR OTHER RANGE ACTIVE
000650  100766                            BMI     CHNERR          ; IF MI, YES - REJECT TRANSITION
000652  052710  100000                    BIS     #100000,(R0)    ; MARK RANGE AS ACTIVE
000656  000207                            RETURN
```

```
                        .TITLE  PRSOBJ - PARSE OBJECTS ACTION ROUTINES
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - PARSE ACTION ROUTINES ASSOCIATED WITH OBJECTS
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                                .SBTTL  .PRNCT - STORE NUMBER OF CONTROLLER TABLES SPECIFIED
                                        ;+
                                        ; *** - .PRNCT - STORE NUMBER OF CONTROLLER TABLES SPECIFIED
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = NUMBER OF CONTROLLER TABLE TO BE SET IN PROCESS SPACE
                                        ;
                                        ; OUTPUT:
                                        ;       NUMBER IS VALIDATED AND THE TRANSITION IS ACCEPTED/REJECTED
                                        ;
                                        ;-

000116                                  .PRNCT::
000116  116767  000000G 000000G                 MOVB    .PNUMB,$NCT             ; STORE THE CONTROLLER COUNT
000124  026727  000000G 000000G                 CMP     .PNUMB,#$MXNCT          ; VALUE IN RANGE ?
000132  101405                                  BLOS    10$                     ; IF LOS, YES
000134  062716  000002                          ADD     #2,(SP)                 ; ELSE, REJECT
000140  012767  000000G 000000G                 MOV     #ERRILN,$ERROR          ; INVALID NUMBER
000146  000207                          10$:    RETURN


        000001                                  .END
```

```
                                                .SBTTL  FILE PARSING ACTION ROUTINES
                                        ;+
                                        ; ** .RNCLF - CLEAR TARGET FILE ADDRESS
                                        ;
                                        ; INPUTS: NONE.
                                        ;
                                        ; OUTPUT: THE TARGET FILE ADDRESS IS CLEARED.
                                        ;
                                        ;-

 000736                                 .RNCLF::
 000736  005067  000000G                        CLR     .RNFSP          ; CLEAR TARGET FILE ADDRESS
 000742  000207                                 RETURN

                                        ;+
                                        ; ** CKFSP - check parsed filespec string
                                        ;
                                        ; Inputs:       .TPARS variables.
                                        ;
                                        ; Output:       F.DEV(.RNFSP)= parsed filespec string
                                        ;               The transition is accepted/rejected
                                        ;-
 000744                                 CKFSP::
 000744  026727  000000G 000036                 CMP     .PSTCN,#MXFSP   ; Check length
 000752  101366                                 BHI     REJ1            ; If HI, too long
                                                                        ; Check for UIC, 9 or less chrs in "[]"
 000754  016701  000000G                        MOV     .PSTPT,R1       ; Address of string
 000760  016700  000000G                        MOV     .PSTCN,R0       ; Byte count of string
 000764  122721  000133         5$:             CMPB    #'[,(R1)+       ; Check for UIC start
 000770  001402                                 BEQ     10$             ; found it, go check UIC
 000772  077004                                 SOB     R0,5$           ; else, cycle
 000774  000412                                 BR      20$             ; no uic is ok, continue
 000776  020027  000012         10$:            CMP     R0,#10.         ; got more than 10. chrs left?
 001002  003402                                 BLE     15$             ; if not, use current count
 001004  012700  000012                         MOV     #10.,R0         ; else, only look at 10. chrs
 001010  122721  000135         15$:            CMPB    #'],(R1)+       ; Check for UIC end
 001014  001402                                 BEQ     20$             ; found it in time-- continue
 001016  077004                                 SOB     R0,15$          ; else, cycle some more
 001020  000743                                 BR      REJ1            ; no "]" within range, reject transition
                                                                        ; continue
 001022  016702  000000G        20$:            MOV     .RNFSP,R2       ; Point to parsed data area
 001026  062702  000000                         ADD     #F.DEV-1,R2     ; ...
 001032  000167  000000                         CALLR   MOVSTR          ; Move it to parsed data area


                                        ;+
                                        ; ** MOVSTR - move string
                                        ;
                                        ; Inputs:       .PSTCN  = length
                                        ;               .PSTPT  = string address
                                        ;               R2      = target
                                        ;
                                        ; Output:       counted string moved to (R2).
                                        ;
                                        ;               R0, R1, and R2 are clobbered.
                                        ;-
 001036                                 MOVSTR:
```

```
                                        .SBTTL  .SYCBN - SET NUMBER OF CCB'S
                               ;+
                               ; *** - .SYCBN - SET NUMBER OF CCB'S
                               ;
                               ; INPUT:
                               ;        .PNUMB - CCB NUMBER
                               ;        .PNUMH   CCB NUMBER (HIGH ORDER)
                               ;
                               ; OUTPUT:
                               ;        $CCBNM = THE NUMBER OF CCB'S
                               ;        THE TRANSITION IS ACCEPTED/REJECTED.
                               ;
                               ;-

000156                         .SYCBN::
000156  016767  000000G 000000G         MOV     .PNUMB,$CCBNM   ; STORE THE NUMBER OF CCB'S
000164  005767  000000G                 TST     .PNUMH          ; IS IT A WORD VALUE ?
000170  001010                          BNE     5$              ; IF NE, NO .. REJECT
000172  022767  000004  000000G         CMP     #4.,.PNUMB      ; THE NUMBER OF CCB'S MUST X WHERE:
000200  003004                          BGT     5$              ; 4. <= X <= 200.
000202  022767  000310  000000G         CMP     #200.,.PNUMB    ;       ...
000210  002005                          BGE     10$             ;       ...
000212  062716  000002         5$:      ADD     #2,(SP)         ; ELSE REJECT THE TRANSITION
000216  012767  000000G 000000G         MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000224  000207                 10$:     RETURN
```

```
                                                .SBTTL  $PUXPR - PURGE MODULE X25-PROTOCOL
                                        ;+
                                        ; *** - $PUXPR - PURGE MODULE X25-PROTOCOL
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = REQUEST CODE
                                        ;
                                        ; OUTPUT:
                                        ;       THE REQUESTED DTE OR GROUP IS PURGED.
                                        ;
                                        ;-
 000500                                 $PUXPR::
 000500  004767  001336                         CALL    TSTPSI          ; MAKE SURE THIS IS A PSI SYSTEM
 000504  032767  000000C 000000G                BIT     #XP.KND!XP.DTE,$OPRND ; PURGE DTES ?
 000512  001402                                 BEQ     10$             ; IF EQ, NOT DTES
 000514                                         CALLR   PURDTE          ; PURGE DTES
 000520  032767  000000C 000000G        10$:    BIT     #XP.KNG!XP.CUG,$OPRND ; PURGE GROUPS ?
 000526  001402                                 BEQ     20$             ; IGNORE PURGE CHARACTERISTICS
 000530                                         CALLR   PURCUG          ; PURGE GROUPS
 000534  000207                         20$:    RETURN
```

```
                                                                      B  7
PURMOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE               REFERENCES

CHKDTE    001004 R           7-171      #8-216
CS.CHN  = ******  GX         7-181       8-221
CS.CUG  = ******  GX         7-189       8-223       9-256
CS.DSA  = ******  GX        10-349
CS.DSC  = ******  GX        10-354
CS.DST  = ******  GX        10-317
CS.DTE  = ******  GX         7-151
CS.LLC  = ******  GX        11-391
CS.MOD  = ******  GX        11-396
CS.PVC  = ******  GX         7-185       8-225
CS.RDT  = ******  GX        13-447
CS.X29  = ******  GX        12-417
C$HLEN  = ******  GX         7-180
C$LENX  = ******  GX         7-183       9-265
C$ULEN  = ******  GX         7-188       9-266
C$UNAM  = ******  GX         9-260       9-262
C.STS   = ******  GX         7-151       7-181       7-185       7-189       8-221       8-223       8-225       9-256       10-317
                            10-349      10-354      11-391     *11-396      12-417      13-447
DEALC     002064 R           7-178       9-267      10-348      10-353      10-358      13-458     #15-510
D$ADLN  = ******  GX        10-351
D$ALEN  = ******  GX        10-352
D$CLEN  = ******  GX        10-357
D$CVLN  = ******  GX        10-356
D$FLNX  = ******  GX        10-346
D$SLEN  = ******  GX        10-347
D$SNAM  = ******  GX        10-323      10-325      10-338      10-340
D$STYP  = ******  GX        10-319
D$TADR  = ******  GX         7-156       8-232
D$TCT   = ******  GX        *7-166
D$THSH  = ******  GX         7-173
D$TLEN  = ******  GX         7-177
ERRDLM    000354 R          #5-102       8-230
ERRDST    000432 R          #5-103      10-371
ERRNDT    000067 R          #5-97        7-201
ERRNSD    000016 R          #5-96       10-373      13-469
ERRNSG    000130 R          #5-98        9-278
ERRPSI    000313 R          #5-101      14-486
ERRXSV    000173 R          #5-99       11-401
ERRX29    000243 R          #5-100      12-424
EX$ERR  = 000002            #5-95        5-96        5-97        5-98        5-99        5-100       5-101       5-102       5-103
G$LENX  = ******  GX         7-187
L$LCTM  = ******  GX       *11-395
L$LENX  = ******  GX         7-176
L$LNAM  = ******  GX        11-393
PURCUG    001120 R           6-127      #9-250
PURDTE    000536 R           6-124      #7-141
P$VLEN  = ******  GX         7-184
P$VOWN  = ******  GX         8-227
R$DLEN  = ******  GX        13-457
R$DNAM  = ******  GX        13-451      13-453
R$LENX  = ******  GX        13-456
SPCDST    000000 R          #5-88       10-332
```

```
                                                                      C  7
PURMOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 2
```

PRGFIL - PURGE SERVICE FILE BLOCKS

```
                                           .SBTTL  PRGFIL - PURGE SERVICE FILE BLOCKS
                                  ;+
                                  ; ** PRGFIL - PURGE SERVICE FILE BLOCKS
                                  ;
                                  ; INPUTS:        $OPRND = PARSED OPTIONS
                                  ;
                                  ; OUTPUT:        THE SPECIFIED FILE(S) ARE PURGED.
                                  ;
                                  ; REGISTERS:     R5 IS PRESERVED.
                                  ;-

 000522                           PRGFIL: SAVRG   R4                      ; SAVE NODE ADDRESS
 000524 012704  000566'                   MOV     #FTBL,R4                ; POINT TO CONTROL TABLE
 000530 012400                    10$:    MOV     (R4)+,R0                ; PICK UP NEXT OFFSET
 000532 001413                            BEQ     20$                     ; IF EQ, DONE
 000534 032467  000000G                   BIT     (R4)+,$OPRND            ; PURGE THIS FILE ?
 000540 001773                            BEQ     10$                     ; NOPE
 000542 066700  177236                    ADD     FILPT,R0                ; POINT TO FILE NAME BLOCK ADDRESS
 000546 011002                            MOV     (R0),R2                 ; GET NAME BLOCK ADDRESS
 000550 001767                            BEQ     10$                     ; IF EQ, NO FILE TO PURGE !
 000552 005010                            CLR     (R0)                    ; DISASSOCIATE NAME WITH NODE
 000554 004767  000034                    CALL    DEAFIL                  ; TRY TO DEALLOCATE FILE NAME BLOCK
 000560 000763                            BR      10$                     ; CONTINUE

 000562                           20$:    RESRG   R4                      ; RESTORE NODE ADDRESS
 000564 000207                            RETURN

 000566                           FTBL:
 000566 000004  000000G                   .WORD   F$SEC,  RN.SLO  ; Secondary loader file name
 000572 000006  000000G                   .WORD   F$TER,  RN.TLO  ; Tertiary loader file name
 000576 000010  000000G                   .WORD   F$LD,   RN.LOD  ; Load file name
 000602 000012  000000G                   .WORD   F$DIA,  RN.DIA  ; Diagnostics file name
 000606 000014  000000G                   .WORD   F$DUM,  RN.DMP  ; Dump file name
 000612 000000                            .WORD   0

```

```
    215
    216 000412  032767  000000G 000000G 40$:     BIT     #OS.THR,$OPRND   ; NEW THRESHOLD VALUE ?
    217 000420  001403                             BEQ     50$              ; IF EQ, NO
    218 000422  016765  000000G 000020             MOV     $THRSH,C.THSH(R5) ; UPDATE THE THRESHOLD VALUE
    219 000430  000405                     50$:    BR      105$             ; RETURN TO CALLER
    220                                    ;
    221                                    ; ERROR CONDITIONS
    222                                    ;
    223 000432                             101$:   ERROR$  #ERRBUF          ; CANNOT FIND BUFFER TEMPLATE
    224 000442  000261                             SEC                      ; INDICATE FAILURE
    225 000444  012605                     105$:   MOV     (SP)+,R5         ; RESTORE TPARS REGISTER
    226 000446  000207                             RETURN                   ; RETURN TO CALLER
    227
    228         000001                             .END
```

B 10

```
SETCNT      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 2
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-75
ERROR$            #4-64         5-172        5-184
PRINT             #4-59
```

C 10

```
**FILE**ID**SETCUG
```

Table of contents

```
     5-   82    LOCAL DATA
     6-   99    $SEDTE - SET LOCAL DTE PARAMETERS
     7-  194    INSDTE - INSERT DTE TEMPLATE IN LIST
     8-  229    SETLIN - SET LINE ID IN DTE TEMPLATE
     9-  264    SETCHN - CREATE CHANNEL LIST
    10-  327    DSTATE - CHANGE DTE STATE
```

```
SETDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE              REFERENCES

$DTE    = ******  GX       6-123     6-153
$DVNAM  = ******  GX       8-248
$EDMSG  = ******  GX       10-367
$FLAGS  = ******  GX       8-253     8-256
$MCI    = ******  GX       6-164     6-170
$OPRND  = ******  GX       6-121     6-135     6-151     6-181
$PSIPT  = ******  GX       7-207
$RLBF   = ******  GX       9-294
$RQBF   = ******  GX       6-141     9-307
$SEDTE    000134 RG       #6-112
$SWICH  = ******  GX       6-137     6-159     6-162     6-171     6-174     6-177     10-343    10-372
$TMLST  = ******  GX       6-116    *7-224    *9-291    *9-311     10-345
$XXAVL  = ******  GX       6-139     9-293     9-305
```

```
SETDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 3
```

B 13

```
SETLOG  - SET LOGGING PARAMETER MACRO V05.03b Saturday 29-Jun-85 05:21  Page 9-1
Symbol table

CS.EVT= ****** GX       EX$WAR= 000000          F.ADD   000016          OM.CON= ****** GX       $CFERR= ****** GX
CS.LOG= ****** GX       FF.ADD= 000020          F.CEV   000030          OM.EVE= ****** GX       $CLASS= ****** GX
CS.MOD= ****** GX       FF.CIR= 000040          F.CIR   000024          OM.KEV= ****** GX       $EVENT= ****** GX
C.CLS   000004          FF.CON= 000001          F.CLS   000002          OM.MON= ****** GX       $KNEVT= ****** GX
C.EVT   000006          FF.FIL= 000002          F.EVT   000004          OM.STA= ****** GX       $OPRND= ****** GX
C.LEN = 000020          FF.HST= 040000          F.FLG   000014          PROEVT  000774R         $QINSN= ****** GX
C.LEN1= 000005          FF.LIN= 000010          F.LEN   000016          SNKTBL  000000R         $RQBF = ****** GX
C.SNK   000016          FF.MOD= 000100          F.LENX= ****** GX       SW.LON= ****** GX       $SAVAL= ****** GX
C.STA   000004          FF.MON= 000004          F.LIN   000020          TMPLOC= 000004          $SELOG  000160RG
C.STS = ****** GX       FF.MSK= 000077          F.LNK   000000          WC.EVE= ****** GX       $SWICH= ****** GX
ERRBUF  000010R         FF.PRT= 000200          F.MOD   000022          WC.LOG= ****** GX       $TMLST= ****** GX
ERREVT  000116R         FF.QUL= 000370          F.REM   000026          $BYTMN= ****** GX       $WILD = ****** GX
ERRLOG  000052R         FF.REM= 100000          F.SEV   000004          $BYTXT= ****** GX       $XXAVL= ****** GX
EX$ERR= 000002          FINEVT  000730R         L.CUST= 074000

. ABS.  000040    000   (RW,I,GBL,ABS,OVR)
        001070    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9204 Words  ( 36 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.01
SY:SETLOG.V2,[132,134]SETLOG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETLOG
```

C 13

```
SETLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 1
```

```
                                                .SBTTL  SETDSC - SET DSC$DF PARAMETERS
                                        ;+
                                        ; ** SETDSC - SET DSC$DF PARAMETERS
                                        ;
                                        ; INPUTS        R2 = ADDRESS OF DSC$DF TEMPLATE
                                        ;               $OPRND = PARAMETERS SELECTED
                                        ;
                                        ; OUTPUT        DSC$DF TEMPLATE UPDATED.
                                        ;-

001402                                  SETDSC:
001402  032767  000000G 000000G                 BIT     #XS.CLM,$OPRND  ; CALL MASK PRESENT?
001410  001440                                  BEQ     20$             ; IF EQ, NO
001412  016701  177776G                         MOV     $CLV-2,R1       ; GET LENGTH OF PARSED VALUE
001416  032767  000000G 000000G                 BIT     #XS.CLV,$OPRND  ; VALUE PRESENT ON COMMAND LINE
001424  001006                                  BNE     5$              ; IF NE YES, COMPARE
001426  032762  000000G 000000G                 BIT     #CS.VAL,C.STS(R2) ; CALL VALUE PRESENT IN TEMPLATE ?
001434  001515                                  BEQ     102$            ; IF EQ, NO - PARAMETER MISSING
001436  116201  000000G                         MOVB    D$CVLN(R2),R1   ; GET LENGTH OF TEMPLATE VALUE
001442  120167  177776G                 5$:     CMPB    R1,$CLM-2       ; DO LENGTHS MATCH
001446  001103                                  BNE     101$            ; IF NE, NO
001450  116762  177776G 000000G                 MOVB    $CLM-2,D$CMLN(R2) ; SAVE LENGTH
001456  012700  000000G                         MOV     #$CLM,R0        ; POINT TO PARSED VALUE
001462  010201                                  MOV     R2,R1           ; COPY ADDRESS
001464  062701  000000G                         ADD     #D$CMSK,R1      ; POINT TO TARGET
001470  012746  000013                          MOV     #11.,-(SP)      ; WORD COUNT
001474  012021                          10$:    MOV     (R0)+,(R1)+     ; MOVE A WORD
001476  005316                                  DEC     (SP)            ; DECREMENT
001500  003375                                  BGT     10$             ; UNTIL DONE
001502  005726                                  TST     (SP)+           ; CLEAN STACK
001504  052762  000000G 000000G                 BIS     #CS.MSK,C.STS(R2) ; REMEMBER OPTION

001512  032767  000000G 000000G 20$:            BIT     #XS.CLV,$OPRND  ; CALL VALUE PRESENT?
001520  001455                                  BEQ     40$             ; IF EQ, NO
001522  016701  177776G                         MOV     $CLM-2,R1       ; GET LENGTH OF PARSED VALUE
001526  032767  000000G 000000G                 BIT     #XS.CLM,$OPRND  ; MASK PRESENT ON COMMAND LINE
001534  001006                                  BNE     25$             ; IF NE YES, COMPARE
001536  032762  000000G 000000G                 BIT     #CS.MSK,C.STS(R2) ; CALL MASK PRESENT IN TEMPLATE ?
001544  001456                                  BEQ     103$            ; IF EQ, NO - PARAMETER MISSING
001546  116201  000000G                         MOVB    D$CMLN(R2),R1   ; GET LENGTH OF TEMPLATE VALUE
001552  120167  177776G                 25$:    CMPB    R1,$CLV-2       ; DO LENGTHS MATCH
001556  001037                                  BNE     101$            ; IF NE, NO
001560  116200  000000G                         MOVB    D$CVLN(R2),R0   ; GET LENGTH OF CALL VALUE
001564  005100                                  COM     R0              ; MAKE IT NEGATIVE
001566  066700  177776G                         ADD     $CLV-2,R0       ; GET DIFFERENCE
001572  062700  000003                          ADD     #3,R0           ; ROUND UP TO 4 BYTE BOUNDARY
001576  042700  000003                          BIC     #3,R0           ; ...
001602  060067  000000G                         ADD     R0,$BYTMN       ; UPDATE BYTE MINIMUM
001606  060067  000000G                         ADD     R0,$BYTXT       ; UPDATE BYTE ALLOCATION
001612  116762  177776G 000000G                 MOVB    $CLV-2,D$CVLN(R2) ; SAVE LENGTH
001620  012700  000000G                         MOV     #$CLV,R0        ; POINT TO PARSED VALUE
001624  010201                                  MOV     R2,R1           ; COPY ADDRESS
001626  062701  000000G                         ADD     #D$CVAL,R1      ; POINT TO TARGET
001632  012746  000013                          MOV     #11.,-(SP)      ; WORD COUNT
001636  012021                          30$:    MOV     (R0)+,(R1)+     ; MOVE A WORD
001640  005316                                  DEC     (SP)            ; DECREMENT
001642  003375                                  BGT     30$             ; UNTIL DONE
```

```
                                                                B 15
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE            REFERENCES

                          5-102      5-103      5-104
L$LCTM  = ******  GX     *10-362
L$LNAM  = ******  GX      10-358
P$SPOR  = ******  GX      10-348    *10-352
RQCB      003012 R        13-464     14-540    #16-604
R$DADR  = ******  GX      14-556
R$DLEN  = ******  GX      14-539
R$DNAM  = ******  GX      14-525     14-527    *14-546    *14-547
R$LENX  = ******  GX      14-542     14-543
SETDSA    001170 R        6-171     #8-227
SETDSC    001402 R        6-176     #9-273
SETDST    001066 R        6-166     #7-200
TSTPSI    002770 R        6-133      14-513    #15-582
XA.KND  = ******  GX      14-523     14-534     14-560
XA.NUM  = ******  GX      14-536     14-552
XSVCHA    001714 R        6-127     #10-342
XS.CLM  = ******  GX      9-274      9-298
XS.CLV  = ******  GX      9-277      9-295
XS.CTM  = ******  GX      6-134      10-360     11-386
XS.CUG  = ******  GX      8-228
XS.DST  = ******  GX      13-451
XS.DTE  = ******  GX      8-233
XS.KND  = ******  GX      6-149      6-159      6-177
XS.MXC  = ******  GX      6-134      10-343     11-389
XS.OBI  = ******  GX      7-208      13-453
XS.OBJ  = ******  GX      7-204      13-453
XS.PRI  = ******  GX      7-201      13-455     13-457
XS.SUB  = ******  GX      8-251
X$9CT   = ******  GX     *11-388
X$9MXC  = ******  GX     *11-391
X29CHA    002054 R        6-132     #11-380
$BYTMN  = ******  GX     *8-241     *9-310     *10-350    *13-467    *14-542
$BYTXT  = ******  GX     *8-242     *9-311     *10-351    *13-468    *14-543
$CFERR  = ******  GX      6-184      9-326      9-328      9-330      10-368     11-396     13-490     13-492     13-494
                          14-566     14-569     15-585
$CLM    = ******  GX      9-282      9-284      9-285      9-297
$CLV    = ******  GX      9-276      9-303      9-307      9-312      9-313
$CT     = ******  GX      10-362     11-388
$CUG    = ******  GX      8-230      8-231
$DST    = ******  GX      6-151      6-153      13-482     13-483     14-525     14-527     14-546     14-547
$DTE    = ******  GX      8-235      8-244      14-554
$MCI    = ******  GX     *10-346     10-347     10-352     11-391
$OBJ    = ******  GX      7-206      7-210      7-211
$OPRND  = ******  GX      6-134      6-149      6-159      6-177      7-201      7-204      7-208      8-228      8-233
                          8-251      9-274      9-277      9-295      9-298      10-343     10-360     11-386     11-389
                          13-451     13-453     13-455    *13-457     14-523     14-534     14-536     14-552     14-560
$PRI    = ******  GX      7-203     *13-458
$PSIPT  = ******  GX      6-140      10-345     14-514     15-583
$RQBF   = ******  GX      16-605
$SEXAC    002472 RG      #14-512
$SEXSV    000634 RG      #6-125
$SEX29    000646 RG      #6-130
```

```
                                                                C 15
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29      PAGE 3
```

```
SETNOD - SET EXECUTOR NODE OPTI MACRO V05.03b Saturday 29-Jun-85 05:22  Page 7-2
Symbol table

CHKADD  000304R          C.HOP    000010          C.STS = ****** GX          UPDREM  000354R          $NTADD= ****** GX
CS.MOD= ****** GX        C.HOS    000054          ERRNOD  000000R            $CFERR= ****** GX        $NTNAM= ****** GX
CS.NOD= ****** GX        C.LEN =  000056          EX$ERR= 000002             $HOST  = ****** GX       $OPRND= ****** GX
CS.REM= ****** GX        C.NAM    000004          OE.ADD= ****** GX          $ID    = ****** GX       $SAVAL= ****** GX
CS.ROU= ****** GX        C.NID    000010          OE.HOS= ****** GX          $MVASC= ****** GX        $SAVRG= ****** GX
C.ADD   000004           C.NUM    000052          OE.ID = ****** GX          $NODAD= ****** GX        $SENOD  000034RG
C.CST   000006           C.RADD   000010          OE.NAM= ****** GX          $NODNA= ****** GX        $TMLST= ****** GX


. ABS.  000056    000   (RW,I,GBL,ABS,OVR)
        000534    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8410 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.56
SY:SETNOD.V2,[132,134]SETNOD/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETNOD
```

```
SETNOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22      PAGE 1
```

```
                                                .SBTTL  .MODST - PARSE DESTINATION ID
                                        ;+
                                        ; ** .MODST - PARSE DESTINATION ID
                                        ;
                                        ; INPUTS        .TPARS VARIABLES.
                                        ;
                                        ; OUTPUT        $DST = PARSED DESTINATION ID
                                        ;               THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-
000660                                  .MODST::
000660  005067  000002G                         CLR     $DST+2          ; CLEAR THE SECOND HALF OF THE DESTINATION
000664  016700  000000G                         MOV     .PSTPT,R0       ; POINT R0 AT DESTINATION
000670  005001                                  CLR     R1              ; PERIODS NOT ACCEPTABLE
000672  004767  000000G                         CALL    $CAT5           ; CONVERT DESTINATION TO RAD50
000676  010167  000000G                         MOV     R1,$DST         ; STORE THE DESTINATION
000702  103413                                  BCS     10$             ; IF CS, LESS THAN 3 CHARS CONVERTED
000704  005001                                  CLR     R1              ; PERIODS NOT ALLOWED
000706  004767  000000G                         CALL    $CAT5           ; CONVERT REST OF DESTINATION TO RAD50
000712  010167  000002G                         MOV     R1,$DST+2       ; STORE SECOND HALF OF DESTINATION
000716  022767  000006  000000G                 CMP     #6,.PSTCN       ; IS THE DESTINATION TOO BIG ?
000724  103002                                  BHIS    10$             ; IF HIS, NO - OKAY
000726  000261                                  SEC                     ; ELSE, INDICATE FAILURE
000730  000401                                  BR      20$             ; AND LEAVE
000732  000241                          10$:    CLC                     ; INDICATE SUCCESS
000734  000207                          20$:    RETURN
```

```
;****
; LOCAL MACROS
;****

        .MACRO  PRINT   TEXT
        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
        CALL    $PRINT          ; PRINT MESSAGE
        .ENDM   PRINT

```

C 3

```
PRSPRO - ACTION ROUTINES FOR PR MACRO V05.03b Saturday 29-Jun-85 05:19  Page 6-1
Symbol table

ERRILN= ****** GX       $ERROR= ****** GX       $NCT   = ****** GX      .PNUMB= ****** GX       .PRNLT  000064RG
ERRPRC= ****** GX       $FNPRO= ****** GX       $NLT   = ****** GX      .PRNAM  000000RG        .PSTCN= ****** GX
ERRPRO= ****** GX       $MXNCT= ****** GX       $PRNAM= ****** GX      .PRNCT  000116RG        .PSTPT= ****** GX
$CAT5 = ****** GX       $MXNLT= ****** GX       $TMLST= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000150    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 51 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:04.18
SY:PRSPRO.V2,[132,134]PRSPRO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSPRO
```

D 3

```
PRSPRO      CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 1
```

```
    433 001036  016700  000000G                 MOV     .PSTCN,R0       ; Get string length
    434 001042  016701  000000G                 MOV     .PSTPT,R1       ; Get string address
    435 001046  110022                          MOVB    R0,(R2)+        ; Stuff length
    436 001050  112122                  10$:    MOVB    (R1)+,(R2)+     ; MOVE...
    437 001052  077002                          SOB     R0,10$          ; ...IT
    438 001054  000207                          RETURN
    439
    440         000001                          .END
```

```
                                                .SBTTL  .SYRBN - SET NUMBER OF RDB'S
                                        ;+
                                        ; *** - .SYRBN - SET NUMBER OF RDB'S
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = NUMBER OF RDB'S
                                        ;       .PNUMH = NUMBER OF RDB'S (HIGH ORDER)
                                        ;
                                        ; OUTPUT:
                                        ;       $RDBNM = THE NUMBER OF RDB'S
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-

000226                                  .SYRBN::
000226  016767  000000G 000000G                 MOV     .PNUMB,$RDBNM   ; STORE THE NUMBER OF RDB'S
000234  005767  000000G                         TST     .PNUMH          ; IS IT A WORD VALUE ?
000240  001004                                  BNE     5$              ; IF NE, NO .. REJECT
000242  022767  000004  000000G                 CMP     #4.,.PNUMB      ; THE NUMBER OF RDB'S MUST BE X WHERE:
000250  101405                                  BLOS    10$             ; IF LOS, YES - OK
                                        ;;;     BGT     5$              ; 4. <= X <= 200.
                                        ;;;     CMP     #200.,.PNUMB    ;       ...
                                        ;;;     BGE     10$             ;       ...
000252  062716  000002                  5$:     ADD     #2,(SP)         ; ELSE REJECT THE TRANSITION
000256  012767  000000G 000000G                 MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000264  000207                          10$:    RETURN
```

```
    131                                           .SBTTL  PURDTE - PURGE DTE DESCRIPTOR
    132                                 ;+
    133                                 ; ** PURDTE - PURGE DTE DESCRIPTOR
    134                                 ;
    135                                 ; INPUTS        $OPRND = REQUEST TYPE
    136                                 ;
    137                                 ; OUTPUT        THE REQUESTED DTE(S) ARE PURGED, ALONG
    138                                 ;               WITH ANY ASSOCIATED PVCS, CHANNELS AND CUGS.
    139                                 ;
    140                                 ;-
    141 000536                          PURDTE:
    142 000536  005004                            CLR     R4                      ; NONE FOUND YET
    143 000540  012702  000000G                   MOV     #$TMLST,R2              ; POINT TO LISTHEAD
    144                                 ;
    145                                 ; FIND THE DTE DESCRIPTOR
    146                                 ;
    147
    148 000544  010205                  10$:      MOV     R2,R5                   ; SAVE PREVIOUS ADDRESS
    149 000546  011202                            MOV     (R2),R2                 ; NEXT ADDRESS
    150 000550  001501                            BEQ     40$                     ; BR IF END OF LIST
    151 000552  122762  000000G 000000G           CMPB    #CS.DTE,C.STS(R2)       ; CHECK FOR DTE BLOCK
    152 000560  001371                            BNE     10$                     ; BR UNLESS DTE
    153 000562  032767  000000G 000000G           BIT     #XP.KND,$OPRND          ; CHECK FOR KNOWN DTES
    154 000570  001013                            BNE     30$                     ; BR IF WILDCARD
    155 000572  010200                            MOV     R2,R0                   ; COPY TEMPLATE ADDRESS
    156 000574  062700  000000G                   ADD     #D$TADR,R0              ; POINT TO DTE ADDRESS
    157 000600  012701  000000G                   MOV     #$DTE,R1                ; POINT TO PARSED DTE
    158 000604  012703  000005                    MOV     #5,R3                   ; LENGTH IN WORDS
    159 000610  022021                  20$:      CMP     (R0)+,(R1)+             ; COMPATE STRINGS
    160 000612  001354                            BNE     10$                     ; BRANCH IF WRONG DTE
    161 000614  005303                            DEC     R3                      ; REDUCE COUNT
    162 000616  003374                            BGT     20$                     ; BR IF MORE
    163 000620  010604                  30$:      MOV     SP,R4                   ; INDICATE WILDCARDS OR FOUND
    164 000622  032767  000000G 000000G           BIT     #XP.CTM,$OPRND          ; PURGE COUNTER TIMER ?
    165 000630  001403                            BEQ     35$                     ; IF EQ, PURGE EVERYTHING
    166 000632  005062  000000G                   CLR     D$TCT(R2)               ; NO MORE COUNTER TIMER
    167 000636  000742                            BR      10$                     ; CONTINUE
    168                                 ;
    169                                 ; PURGE DTE, CHANNELS, PVCS AND CUGS
    170                                 ;
    171 000640  004767  000140          35$:      CALL    CHKDTE                  ; MAKE SURE DTE CAN BE DELETED
    172 000644  103737                            BCS     10$                     ; IF CS, TRY NEXT
    173 000646  016200  000000G                   MOV     D$THSH(R2),R0           ; GET NUMBER OF HASH TABLE ENTRIES
    174 000652  006300                            ASL     R0                      ; 2 WORDS OF POOL.. PER ENTRY
    175 000654  006300                            ASL     R0                      ; CONVERT TO BYTES
    176 000656  062700  000000G                   ADD     #L$LENX,R0              ; ADD SIZE OF FIXED PART OF DTE DESCRIPTOR
    177 000662  012701  000000G                   MOV     #D$TLEN,R1              ; LENGTH OF A DTE TEMPLATE
    178 000666  004767  001172          36$:      CALL    DEALC                   ; DEALLOCATE IT
    179 000672  005000                            CLR     R0                      ; CHANNELS INCLUDED IN HASH TABLE
    180 000674  012701  000000G                   MOV     #C$HLEN,R1              ; LENGTH OF CHANNEL BLOCK
    181 000700  122762  000000G 000000G           CMPB    #CS.CHN,C.STS(R2)       ; CHANNEL LIST ?
    182 000706  001767                            BEQ     36$                     ; IF EQ, RUN DOWN CHANNEL LIST
    183 000710  012700  000000G                   MOV     #C$LENX+X$LENX,R0               ; LENGTH OF CIRCUIT BLOCK IN POOL..
    184 000714  012701  000000G                   MOV     #P$VLEN,R1              ; LENGTH OF A PVC DESCRIPTOR
    185 000720  122762  000000G 000000G           CMPB    #CS.PVC,C.STS(R2)       ; PVC ?
    186 000726  001757                            BEQ     36$                     ; IF EQ, RETURN PVC BLOCK
    187 000730  012700  000000G                   MOV     #G$LENX,R0              ; LENGTH OF CUG BLOCK IN POOL..
```

```
                                                                        C  7
PURMOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20        PAGE 2
SYMBOL CROSS REFERENCE                                      CREF    04.00

SYMBOL   VALUE              REFERENCES

TSTPS!     002042 R          6-121       10-308      13-441      #14-483
XA.KND   = ******    GX     13-449       13-461
XP.CTM   = ******    GX      7-164
XP.CUG   = ******    GX      6-125
XP.DTE   = ******    GX      6-122
XP.KND   = ******    GX      6-122        7-153       7-193
XP.KNG   = ******    GX      6-125        9-258       9-270
XSVCTM     001572 R         10-302      #11-387
XS.CTM   = ******    GX     10-310
XS.KND   = ******    GX     10-321       10-342      10-362
X$LENX   = ******    GX      7-183
X$9CT    = ******    GX    *12-419
X29CTM     001650 R         10-307      #12-413
$BUFF    = ******    GX      8-229        8-234
$BYTMN   = ******    GX    *15-515
$BYTXT   = ******    GX    *15-516
$CFERR   = ******    GX      7-201        8-234       9-278      10-371      10-373      11-401      12-424      13-469      14-486
$CUG     = ******    GX      9-260        9-262
$DST     = ******    GX     10-323       10-325      13-451      13-453
$DTE     = ******    GX      7-157
$EDMSG   = ******    GX      8-233
$OPRND   = ******    GX      6-122        6-125       7-153       7-164       7-193       9-258       9-270      10-310      10-321
                            10-342       10-362      13-449      13-461
$PSIPT   = ******    GX     14-484
$PUXAC     001710 RG       #13-440
$PUXPR     000500 RG       #6-120
$PUXSV     001246 RG       #10-300
$PUXS9     001260 RG       #10-305
$RLBF    = ******    GX     15-521
$SAVVR   = ******    GX      8-217
$TMLST   = ******    GX      7-143        9-252      10-313      11-388      12-414      13-443     *15-519
$XXAVL   = ******    GX     15-520
.CONER   = ******    GX     10-372
```

```
                                                                        D  7
PURMOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20        PAGE 3
```

```
                                              .SBTTL  DEAFIL - DEALLOCATE A FILE BLOCK
                                      ;+
                                      ; ** DEAFIL - DEALLOCATE A FILE BLOCK
                                      ;
                                      ; INPUTS:         R2 =    ADDRESS OF SERVICE FILE BLOCK
                                      ;
                                      ; OUTPUT:         THE FILE NAME USE COUNT IS REDUCED.  IF THERE
                                      ;                 ARE NO MORE USERS OF THE FILE NAME, THE POOL
                                      ;                 BYTE EXTENSION ($BYTMN,$BYTXT) IS REDUCED BY
                                      ;                 THE SIZE OF THE RUN-TIME SERVICE FILE BLOCK, AND
                                      ;                 THE CURRENT FILE BLOCK IS REMOVED FROM THE SERVICE FILE
                                      ;                 LIST AND RETURNED TO THE FREE LIST.
                                      ;
                                      ;                 R3, R4 AND R5 ARE PRESERVED.
                                      ;-
 000614  005362  000002               DEAFIL: DEC     F.CNT(R2)               ; DECREMENT USE COUNT
 000620  001030                               BNE     20$                     ; IF NE, DONE

 000622  116200  000004                       MOVB    F.DEV-1(R2),R0          ; GET DEVICE LENGTH
 000626  062700  000011                       ADD     #6+3,R0                 ; COUNT FIXED OVERHEAD AND ROUND UP
 000632  042700  000003                       BIC     #3,R0                   ; ALIGN ON 4 BYTE BOUNDARY
 000636  160067  000000G                      SUB     R0,$BYTXT               ; UPDATE ACTUAL ALLOCATION
 000642  160067  000000G                      SUB     R0,$BYTMN               ; UPDATE MINIMUM ALLOCATION

 000646  012701  000000G                      MOV     #$FILHD,R1              ; POINT TO FILE BLOCK LISTHEAD
 000652  000402                               BR      15$                     ; ...
 000654  011101                       10$:    MOV     (R1),R1                 ; NEXT
 000656  001412                               BEQ     101$                    ; SHOULD NEVER HAPPEN, BUT..
 000660  021102                       15$:    CMP     (R1),R2                 ; IS THIS THE UNLINK POINT ?
 000662  001374                               BNE     10$                     ; NO - KEEP GOING
 000664  011211                               MOV     (R2),(R1)               ; CLOSE UP LIST
 000666  012701  000043                       MOV     #F.LEN1,R1              ; LENGTH OF FILE BLOCK
 000672  012700  000000G                      MOV     #$XXAVL,R0              ; POINT TO FREE LIST
 000676  004767  000000G                      CALL    $RLBF                   ; RELEASE CORE BLOCK
 000702  000207                       20$:    RETURN

 000704  000167  000000G              101$:   JMP     .CONER                  ; OOPS
```

```
                                                                       C 9
SETBUF - SET BUFFER PARAMETERS  MACRO V05.03b Saturday 29-Jun-85 05:20  Page 5-2
Symbol table

CH.DUP= 000001        CS.BUF= ****** GX     C.SBSZ  000016        OS.CCB= ****** GX     $CFERR= ****** GX
CH.D11= 000100        CS.MOD= ****** GX     C.STS = ****** GX     OS.RBN= ****** GX     $EDMSG= ****** GX
CH.ECH= 000002        C.CBNM  000004        C.THSH  000020        OS.RBS= ****** GX     $OPRND= ****** GX
CH.MDT= 000020        C.CBSZ  000006        C.UNT   000004        OS.SBN= ****** GX     $RDBNM= ****** GX
CH.PAR= 000100        C.CSR   000012        DDM     000002R       OS.THR= ****** GX     $RDBSZ= ****** GX
CH.PRM= 000200        C.CW0   000006        ERRBUF  000004R       RATETB  000124R       $SDBNM= ****** GX
CH.PRT= 000007        C.CW1   000010        ERRTHS  000041R       RDBSZ   000000R       $SEBUF  000162RG
CH.STA= 000040        C.RBNM  000010        EX$ERR= 000002        $BUFF = ****** GX     $THRSH= ****** GX
CH.SYN= 000004        C.RBSZ  000012        EX$WAR= 000000        $CCBNM= ****** GX     $TMLST= ****** GX
CH.TYP= 000010        C.SBNM  000014

. ABS.  000022    000   (RW,I,GBL,ABS,OVR)
        000450    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10104 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.39
SY:SETBUF.V2,[132,134]SETBUF/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETBUF
```

```
                                                                  D 9
SETBUF      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20       PAGE 1
```

```
**FILE**ID**SETCUG







  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     CCCCCCCC   UU      UU     GGGGGGGG
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     CCCCCCCC   UU      UU     GGGGGGGG
SS           EE               TT       CC           UU      UU   GG
SS           EE               TT       CC           UU      UU   GG
SS           EE               TT       CC           UU      UU   GG
SS           EE               TT       CC           UU      UU   GG
  SSSSSS     EEEEEEEE         TT       CC           UU      UU   GG
  SSSSSS     EEEEEEEE         TT       CC           UU      UU   GG
        SS   EE               TT       CC           UU      UU   GG  GGGGGG
        SS   EE               TT       CC           UU      UU   GG  GGGGGG
        SS   EE               TT       CC           UU      UU   GG      GG   ....
        SS   EE               TT       CC           UU      UU   GG      GG   ....
SSSSSSSS     EEEEEEEEEE       TT         CCCCCCCC   UUUUUUUUUU     GGGGGG     ....
SSSSSSSS     EEEEEEEEEE       TT         CCCCCCCC   UUUUUUUUUU     GGGGGG     ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTT"TTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                        .TITLE  SETDTE - SET LOCAL DTE PARAMETERS
                        .IDENT  /V05.00/
                        .ENABL  LC

                ;
                ; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - SET LOCAL DTE PARAMETERS
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
SETDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 3
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-70
ERROR$            #4-59        6-187       6-189       10-368
PRINT             #4-54
```

```
**FILE**ID**SETLOG
```

```
SETLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE               REFERENCES

CS.EVT   = ******    GX    7-220     7-243     8-277
CS.LOG   = ******    GX    7-177
CS.MOD   = ******    GX    7-179
C.CLS      000004         #6-132     7-222    *7-244     8-279     8-279
C.EVT      000006         #6-133    *7-229    *7-230    *7-231    *7-232    *7-246    *7-247    *7-248    *7-249
C.LEN    = 000020         #6-135     7-238
C.LEN1   = 000005         #6-141
C.SNK      000016         #6-134     7-224    *7-245     8-281     8-281
C.STA      000004         #6-140    *7-182    *7-184
C.STS    = ******    GX    7-177    *7-179     7-220    *7-243     8-277
ERRBUF     000010 R       #5-111     7-241
ERREVT     000116 R       #5-113     7-215
ERRLOG     000052 R       #5-112     7-175
EX$ERR   = 000002         #5-109     5-111     5-112
EX$WAR   = 000000         #5-110     5-113
FF.ADD   = 000020         #4-93      4-93
FF.CIR   = 000040         #4-93      4-93
FF.CON   = 000001         #4-93      5-103     7-157     7-163
FF.FIL   = 000002         #4-93      5-104     7-157     7-166
FF.HST   = 040000         #4-93
FF.LIN   = 000010         #4-93      4-93
FF.MOD   = 000100         #4-93      4-93
FF.MON   = 000004         #4-93      5-105     7-157     7-160
FF.MSK   = 000077         #4-93
FF.PRT   = 000200         #4-93      4-93
FF.QUL   = 000370         #4-93
FF.REM   = 100000         #4-93
FINEVT     000730 R        7-250    #8-273
F.ADD      000016         #4-93
F.CEV      000030         #4-93
F.CIR      000024         #4-93
F.CLS      000002         #4-93
F.EVT      000004         #4-93
F.FLG      000014         #4-93
F.LEN      000016         #4-93
F.LENX   = ******    GX    7-252     7-253
F.LIN      000020         #4-93
F.LNK      000000         #4-93
F.MOD      000022         #4-93
F.REM      000026         #4-93
F.SEV      000004         #4-93
L.CUST   = 074000         #6-122     9-303
OM.CON   = ******    GX    7-164
OM.EVE   = ******    GX    7-188
OM.KEV   = ******    GX    7-199     7-256
OM.MON   = ******    GX    7-161
OM.STA   = ******    GX    7-167
PROEVT     000774 R        7-213    #9-302
SNKTBL     000000 R       #5-103     7-210
SW.LON   = ******    GX    7-180
TMPLOC   = 000004         #6-131     6-139
WC.EVE   = ******    GX    7-190
```

```
SETLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 2
```

```
    320 001644  005726                      TST     (SP)+           ; CLEAN STACK
    321 001646  052762  000000G 000000G     BIS     #CS.VAL,C.STS(R2) ; REMEMBER OPTION
    322 001654  000207              40$:    RETURN
    323                             ;
    324                             ; ERRORS
    325                             ;
    326 001656                      101$:   ERROR$  #ERRLEN         ; CALL MASK AND CALL VALUE LENGTH ERROR
    327 001666  000207                      RETURN
    328 001670                      102$:   ERROR$  #ERRCLV         ; CALL VALUE MISSING
    329 001700  000207                      RETURN
    330 001702                      103$:   ERROR$  #ERRCLM         ; CALL MASK MISSING
    331 001712  000207                      RETURN
```

```
                                                                    C 15
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29        PAGE 3
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL   VALUE               REFERENCES

$SUB1    = ******  GX         8-255
$SUB2    = ******  GX         8-256         8-259
$TMLST   = ******  GX        10-353        11-381     *13-479     *14-550
$XXAVL   = ******  GX        13-460        14-538
.CONER   = ******  GX         6-185
```

```
                                                                    D 15
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29        PAGE 4
```

```
                                                                     C 16
SETNOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22       PAGE 1
SYMBOL CROSS REFERENCE                                      CREF    04.00
SYMBOL   VALUE              REFERENCES

CHKADD     000304 R         #6-213
CS.MOD   = ******   GX       5-157
CS.NOD   = ******   GX       5-155
CS.REM   = ******   GX       7-253       7-281
CS.ROU   = ******   GX       6-221
C.ADD      000004           #4-118       6-223      6-225
C.CST      000005           #4-119
C.HO[illegible]      000010           #4-120
C.HOS      000034           #4-110      *5-189     *5-192     *5-193
C.LEN    = 000050           #4-111
C.NAM      000004           #4-106      *5-160     *5-161
C.NID      000010           #4-107       5-181
C.NUM      000052           #4-109      *5-166      5-167     *5-170     *5-171      5-172
C.RADD     000010           #4-131       7-255     *7-259     *7-260      7-283      7-283
C.STS    = ******   GX       5-155      *5-157      6-221      7-253      7-281
ERRMOD     000000 R         #4-96        5-198
EX$ERR   = 000002           #4-95        4-96
OE.ADD   = ******   GX       5-164
OE.HOS   = ******   GX       5-187
OE.ID    = ******   GX       5-178
OE.NAM   = ******   GX       5-158
UPDREM     000354 R         #7-243
$CFERR   = ******   GX       5-198
$HOST    = ******   GX       5-190       5-193
$ID      = ******   GX       5-182
$MVASC   = ******   GX       5-185
$NODAD   = ******   GX       5-168       5-171      6-225
$NODNA   = ******   GX       5-160       5-161      5-162      5-163
$NTADD   = ******   GX      *5-172
$NTNAM   = ******   GX      *5-162      *5-163
$OPRND   = ******   GX       5-158       5-164      5-178      5-187
$SAVAL   = ******   GX       6-214
$SAVRG   = ******   GX       7-244
$SENOD     000034 RG        #5-148
$TMLST   = ******   GX       5-149       6-215      7-245     *7-270      7-277     *7-290
```

```
                                                                     D 16
SETNOD      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22       PAGE 2
```

```
                                        .SBTTL  .MOCLM  - PARSE CALL MASK
                                        .SBTTL  .MOCLV  - PARSE CALL VALUE
                                ;+
                                ; ** .MOCLM - PARSE CALL MASK
                                ; ** .MOCLV - PARSE CALL VALUE
                                ;
                                ; INPUTS        .TPARS VARAIBLES
                                ;
                                ; OUTPUT        CALL MASK OR VALUE IS STORED IN
                                ;               $CLM, $CLV RESPECTIVELY.
                                ;
                                ;-
                                        .ENABLE LSB
000736                          .MOCLM::
000736  012702  000026G                 MOV     #$CLM+<2*11.>,R2; PARSED AREA ADDRESS
000742  000402                          BR      10$             ; JOIN COMMON PROCESSING

000744                          .MOCLV::
000744  012702  000026G                 MOV     #$CLV+<2*11.>,R2; TARGET ADDRESS

000750  026727  000000G 000040  10$:    CMP     .PSTCN,#32.     ; CHECK SIZE OF PARSED STRING
000756  101030                          BHI     ILLCAL          ; ILLEGAL CALL MASK OR VALUE
000760  016762  000000G 177750          MOV     .PSTCN,-<2*11.>-2(R2) ; STUFF SIZE
000766  010346                          MOV     R3,-(SP)        ; SAVE SCRATCH REGISTER
000770  010203                          MOV     R2,R3           ; COPY ADDRESS
000772  016746  000000G                 MOV     .PSTCN,-(SP)    ; PUSH LENGTH OF STRING
000776  012700  000013                  MOV     #11.,R0         ; INIT COUNTER
001002  005043                  15$:    CLR     -(R3)           ; CLEAR OUT TARGET
001004  005300                          DEC     R0              ; REDUCE COUNT
001006  003375                          BGT     15$             ; UNTIL ALL 32. CHARS ARE BLANKS
001010  016700  000000G                 MOV     .PSTPT,R0       ; POINT TO INPUT STRING
001014  005001                  20$:    CLR     R1              ; DON'T ALLOW PERIODS
001016  004767  000000G                 CALL    $CAT5           ; CONVERT 3 CHARS
001022  010123                          MOV     R1,(R3)+        ; STUFF INTO TARGET BUFFER
001024  162716  000003                  SUB     #3,(SP)         ; REDUCE CHARS REMAINING COUNT
001030  003371                          BGT     20$             ; IF GT, MORE
001032  005726                          TST     (SP)+           ; ELSE CLEAN OFF COUNT
001034  012603                          MOV     (SP)+,R3        ; RESTORE REGISTER
001036  000207                          RETURN                  ; ACCEPT TRANSITION

001040  012767  000000G 000000G ILLCAL: MOV     #ERRCAL,$ERROR  ; ILLEGAL CALL ADDRESS/VALUE
001046  000622                          BR      REJECT

                                        .DSABL  LSB
```

```
     65                                                   .SBTTL  .OBNUM - STORE THE OBJECT NUNMBER
     66                                           ;+
     67                                           ; *** - .OBNUM - STORE THE OBJECT NUMBER
     68                                           ;
     69                                           ; INPUT:
     70                                           ;         .PNUMB = THE OBJECT NUMBER
     71                                           ;         .PNUMH = THE OBJECT NUMBER (HIGH ORDER)
     72                                           ;
     73                                           ; OUTPUT:
     74                                           ;         $TNUM = THE OBJECT NUMBER
     75                                           ;         THE TRANSITION IS ACCEPTED/REJECTED
     76                                           ;
     77                                           ;-
     78
     79 000000                                    .OBNUM::
     80 000000  016767  000000G 000000G                   MOV     .PNUMB,$TNUM      ; STORE THE OBJECT TASK NUMBER
     81 000006  005767  000000G                           TST     .PNUMH            ; IS IT A WORD VALUE ?
     82 000012  001003                                    BNE     10$               ; IF NE, NO .. REJECT
     83 000014  105767  000001G                           TSTB    .PNUMB+1          ; IS IT A BYTE VALUE ?
     84 000020  001405                                    BEQ     20$               ; IF EQ, YES .. OKAY
     85 000022  062716  000002                    10$:    ADD     #2,(SP)           ; REJECT THE TRANSITION
     86 000026  012767  000000G 000000G                   MOV     #ERRILN,$ERROR    ; INVALID NUMBER
     87 000034  000207                            20$:    RETURN
```

```
PRSPRO      CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 1
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL  VALUE              REFERENCES

ERRILN  = ******  GX        5-103       6-123
ERRPRC  = ******  GX        4-80
ERRPRO  = ******  GX        4-82
$CAT5   = ******  GX        4-73
$ERROR  = ******  GX       *4-80       *4-82       *5-103      *6-123
$FNPRO  = ******  GX        4-78
$MXNCT  = ******  GX        6-120
$MXNLT  = ******  GX        5-100
$NCT    = ******  GX       *6-119
$NLT    = ******  GX       *5-99
$PRNAM  = ******  GX       *4-74
$TMLST  = ******  GX        4-77
.PNUMB  = ******  GX        5-99        5-100       6-119       6-120
.PRNAM    000000 RG        #4-70
.PRNCT    000116 RG        #6-118
.PRNLT    000064 RG        #5-98
.PSTCN  = ******  GX        4-75
.PSTPT  = ******  GX        4-71
```

```
**FILE**ID**PRSREM
```

```
                                                                    D  4
PRSREM - PARSE ACTION ROUTINES  MACRO V05.03b Saturday 29-Jun-85 05:19  Page 9-2
Symbol table

CKFSP   000744RG       F.LEN = 000037         T2      000405R        $RNSCI= ****** GX       .RNCLF  000736RG
CS.REM= ****** GX      H     = 177777         WC.CIR= ****** GX      $RNSDV= ****** GX       .RNDCT  000246RG
C.ADD   000010         HEX     000604R        $CAT5 = ****** GX      $RNSPA= ****** GX       .RNDPA  000230RG
C.LEN = 000012         ILLNUM  000310R        $ERROR= ****** GX      $RNXXX= ****** GX       .RNFSP= ****** GX
C.NAM   000004         MOVSTR  001036R        $NODAD= ****** GX      $TMLST= ****** GX       .RNHAR  000424RG
C.STS = ****** GX      MXFSP = 000036         $NODNA= ****** GX      $WILD = ****** GX       .RNHOS  000172RG
D     = 000001         REJ     000316R        $NTADD= ****** GX      .CAT5E= ****** GX       .RNHO1  000156RG
E     = 000000         REJ1    000730R        $NTNAM= ****** GX      .PNUMB= ****** GX       .RNHO2  000166RG
ERRCID= ****** GX      RN.NAM= ****** GX      $OPRND= ****** GX      .PNUMH= ****** GX       .RNIN2  000020RG
ERRILN= ****** GX      SCR     000000R        $RNDCP= ****** GX      .PSTCN= ****** GX       .RNSCI  000324RG
ERRNOD= ****** GX      STR50   000530R        $RNDPA= ****** GX      .PSTPT= ****** GX       .RNSDV  000212RG
ERRPNA= ****** GX      STR50X  000540R        $RNHAR= ****** GX      .RNCHK  000036RG        .RNSPA  000642RG
F.DEV   000001         T1      000362R        $RNHOS= ****** GX

. ABS.  000037    000    (RW,I,GBL,ABS,OVR)
        001056    001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8258 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:10.25
SY:PRSREM.V2,[132,134]PRSREM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSREM
```

```
                                                                    E  4
PRSREM      CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 1
```

```
                                            .SBTTL  .SYRBS - SET SIZE OF EACH RDB
                                    ;+
                                    ; *** - .SYRBS - SET SIZE OF EACH RDB
                                    ;
                                    ; INPUT:
                                    ;       .PNUMB = SIZE OF EACH RDB
                                    ;       .PNUMH = SIZE OF EACH RDB (HIGH ORDER)
                                    ;
                                    ; OUTPUT:
                                    ;       $RDBSZ = THE SIZE OF EACH RDB
                                    ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                    ;
                                    ;-

 000266                             .SYRBS::
 000266  016767  000000G 000000G            MOV     .PNUMB,$RDBSZ   ; STORE THE SIZE OF EACH RDB
 000274  005767  000000G                    TST     .PNUMH          ; IS IT A WORD VALUE ?
 000300  001023                             BNE     5$              ; IF NE, NO .. REJECT
 000302  012701  000105                     MOV     #69.,R1         ; ASSUME PSI-ONLY SYSTEM
 000306  012700  000000G                    MOV     #$TMLST,R0      ; POINT TO TEMPLATE LISTHEAD
 000312  011000                     1$:     MOV     (R0),R0         ; NEXT
 000314  001406                             BEQ     2$              ; IF EQ, END OF LIST
 000316  122760  000000G 000000G            CMPB    #CS.ROU,C.STS(R0) ; IS THIS THE ROUTING TEMPLATE ?
 000324  001372                             BNE     1$              ; IF NE, NO
 000326  012701  000300                     MOV     #192.,R1        ; ELSE DECNET IS PRESENT
 000332  020167  000000G            2$:     CMP     R1,.PNUMB       ; THE RDB SIZE MUST BE X WHERE:
 000336  003004                             BGT     5$              ; 192. (69. FOR PSI ONLY) <= X <= 4054.
 000340  022767  007726  000000G            CMP     #4054.,.PNUMB   ;       ...
 000346  002005                             BGE     10$             ;
 000350  062716  000002             5$:     ADD     #2,(SP)         ; ELSE REJECT THE TRANSITION
 000354  012767  000000G 000000G            MOV     #ERRILN,$ERROR  ; INVALID NUMBER
 000362  000207                     10$:    RETURN
```

```
    000734  012701  000000G                  MOV     #C$ULEN,R1       ; LENGTH OF A CUG DESCRIPTOR
    000740  022762  000000G 000000G          CMP     #CS.CUG,C.STS(R2) ; IS THIS A USER GROUP BLOCK ?
    000746  001747                           BEQ     36$              ; IF EQ, YES - PURGE IT
    000750  010502                           MOV     R5,R2            ; PICK UP SCAN WHERE WE LEFT OFF
    000752  000674                           BR      10$              ; CONTINUE
    000754  032767  000000G 000000G 40$:     BIT     #XP.KND,$OPRND   ; WILDCARD OPERATION ?
    000762  001002                           BNE     50$              ; BR IF YES
    000764  005704                           TST     R4               ; ANYTHING FOUND ?
    000766  001401                           BEQ     101$             ; BR IF ERROR
    000770  000207                   50$:    RETURN
                                     ;
                                     ; ERRORS
                                     ;
    000772                           101$:   ERROR$  #ERRNDT          ; NO SUCH DTE
    001002  000207                           RETURN

```

D 7

```
PURMOD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

CALLR            #4-75      6-124      6-127      10-312     10-371     10-373     11-401     12-424     13-469     14-486
ERROR$           #4-64      7-201      8-234      9-278
PRINT            #4-59
```

E 7

```
**FILE**ID**PURREM
```

```
                                        .SBTTL  QREM - REMOVE ENTRY FROM QUEUE
                                ;+
                                ; *** - QREM - REMOVE ENTRY FROM QUEUE
                                ;
                                ; INPUTS:
                                ;       R5 = PREVIOUS ENTRY IN QUEUE
                                ;       R2 = ENTRY TO REMOVE
                                ;       $TMLST = QUEUE LISTHEAD
                                ;
                                ; OUTPUTS:
                                ;       R5,R2 = PRESERVED
                                ;       ENTRY IS REMOVED FROM QUEUE
                                ;
                                ;-

000710                          QREM:
000710  011215                          MOV     (R2),(R5)       ; CLOSE UP LIST
000712  001002                          BNE     10$             ; IF NE, NOT LAST IN LIST
000714  010567  000002G                 MOV     R5,$TMLST+2     ; SET NEW LAST ENTRY
000720  000207                  10$:    RETURN

        000001                          .END
```

```
                                                                 D  9
SETBUF      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

CS.BUF  = ******  GX         5-175
CS.MOD  = ******  GX         5-180
C.CBNM    000004            #4-139     *5-183
C.CBSZ    000006            #4-140
C.CSR     000012            #4-155
C.CWO     000006            #4-153
C.CW1     000010            #4-154
C.RBNM    000010            #4-141     *5-188      5-205
C.RBSZ    000012            #4-142      5-204      5-206      *5-207
C.SBNM    000014            #4-143      5-212     *5-214
C.SBSZ    000016            #4-144      5-213
C.STS   = ******  GX         5-175     *5-180
C.THSH    000020            #4-145      5-190     *5-192       5-194     *5-218
C.UNT     000004            #4-152
IDM       000002 R          #4-98
ERRBUF    000004 R          #4-107      5-223
ERRTHS    000041 R          #4-108      5-195
EX$ERR  = 000002            #4-106      4-107
EX$WAR  = 000000            #4-105      4-108
OS.CCB  = ******  GX         5-181
OS.RBN  = ******  GX         5-185
OS.RBS  = ******  GX         5-201
OS.SBN  = ******  GX         5-209
OS.THR  = ******  GX         5-216
RATETB    000124 R          #4-115
RDBSZ     000000 R          #4-97      *5-206
$BUFF   = ******  GX         5-196      5-199
$CCBNM  = ******  GX         5-183
$CFERR  = ******  GX         5-199      5-223
$EDMSG  = ******  GX         5-198
$OPRND  = ******  GX         5-181      5-185      5-201       5-209      5-216
$RDBNM  = ******  GX         5-187
$RDBSZ  = ******  GX         5-203      5-207
$SDBNM  = ******  GX         5-211      5-214
$SEBUF    000162 RG         #5-170
$THRSH  = ******  GX         5-218
$TMLST  = ******  GX         5-172
```

```
                                                                 E  9
SETBUF      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 2
```

Table of contents

```
     5-    86     LOCAL DATA
     6-    98     $SECUG - SET CLOSED USER GROUP PARAMETERS
```

```
                        ;****
                        ; LOCAL MACROS
                        ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT

                                .MACRO  ERROR$  TEXT
                        ;
                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                        ; MESSAGE STRING.
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                .ENDC
                                CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                .ENDM   ERROR$

                                .MACRO  EPRINT  TEXT
                        ;
                        ; PRINT TEXT ON ERROR LUN
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                .ENDC
                                CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                .ENDM   EPRINT

        000040                  DEFHSH  = 32.                   ; DEFAULT HASH TABLE SIZE FOR NEW DTES
```

```
**FILE**ID**SETLOG




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   LL             OOOOOO       GGGGGGGG
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   LL             OOOOOO       GGGGGGGG
SS           EE               TT       LL           OO      OO   GG
SS           EE               TT       LL           OO      OO   GG
SS           EE               TT       LL           OO      OO   GG
SS           EE               TT       LL           OO      OO   GG
  SSSSSS     EEEEEEEE         TT       LL           OO      OO   GG
  SSSSSS     EEEEEEEE         TT       LL           OO      OO   GG
        SS   EE               TT       LL           OO      OO   GG  GGGGGG
        SS   EE               TT       LL           OO      OO   GG  GGGGGG
        SS   EE               TT       LL           OO      OO   GG      GG       ....
        SS   EE               TT       LL           OO      OO   GG      GG       ....
SSSSSSSS     EEEEEEEEEE       TT       LLLLLLLLLL     OOOOOO       GGGGGG         ....
SSSSSSSS     EEEEEEEEEE       TT       LLLLLLLLLL     OOOOOO       GGGGGG         ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
SETLOG  - SET LOGGING PARAMETER MACRO V05.03b Saturday 29-Jun-85 05:21
```

```
                                                                   D 13
SETLOG      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

WC.LOG   = ******  GX       7-158     7-208     7-254
$BYTMN   = ******  GX      *7-252
$BYTXT   = ******  GX      *7-253
$CFERR   = ******  GX       7-175     7-215     7-241
$CLASS   = ******  GX       7-193    *7-202     7-222     7-244     9-303     9-306
$EVENT   = ******  GX      *7-204    *7-205    *7-206    *7-207     7-229     7-230     7-231     7-232     7-246
                            7-247     7-248     7-249     9-308
$KNEVT   = ******  GX       7-192     7-201     9-305
$OPRND   = ******  GX       7-161     7-164     7-167     7-188     7-199     7-256
$QINSN   = ******  GX       7-251
$RQBF    = ******  GX       7-239
$SAVAL   = ******  GX       9-302
$SELOG     000160 RG       #7-156
$SWICH   = ******  GX       7-180
$TMLST   = ******  GX       7-172     7-217     8-273
$WILD    = ******  GX       7-158     7-190     7-208     7-254
$XXAVL   = ******  GX       7-237
```

```
                                                                   E 13
SETLOG      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 3
```

```
                                              .SBTTL  XSVCHA - UPDATE X25-SERVER CHARACTERISTICS
                                      ;+
                                      ; ** XSVCHA -  UPDATE X25-SERVER CHARACTERISTICS
                                      ;
                                      ; INPUTS         NONE.
                                      ;
                                      ; OUTPUT         THE X25-SERVER CHARACTERISTICS ARE UPDATED.
                                      ;-

001714                                XSVCHA:
001714  032767  000000G 000000G               BIT     #XS.MXC,$OPRND    ; UPDATE MAX CIRCUITS ?
001722  001420                                BEQ     5$                ; BR IF NO
001724  016700  000000G                       MOV     $PSIPT,R0         ; POINT TO PSN$DF TEMPLATE
001730  005267  000000G                       INC     $MCI              ; ALWAYS FUDGE AN EXTRA ENTRY
001734  016701  000000G                       MOV     $MCI,R1           ; GET NEW VALUE
001740  166001  000000G                       SUB     P$SPOR(R0),R1     ; CALCULATE DIFFERENCE
001744  006301                                ASL     R1                ; IN BYTES
001746  060167  000000G                       ADD     R1,$BYTMN         ; UPDATE POOL BYTE MINIMUM
001752  060167  000000G                       ADD     R1,$BYTXT         ; UPDATE POOL BYTE ALLOCATION
001756  016760  000000G 000000G               MOV     $MCI,P$SPOR(R0)   ; SET MAX CIRCUIT PARAMETER
001764  012700  000000G               5$:     MOV     #$TMLST,R0        ; POINT TO TEMPLATES
001770  011000                        10$:    MOV     (R0),R0           ; NEXT TEMPLATE
001772  001423                                BEQ     101$              ; BR IF ERROR
001774  122760  000000G 000000G               CMPB    #CS.LLC,C.STS(R0) ; CHECK TYPE
002002  001372                                BNE     10$               ; BR IF NOT LLC
002004  026027  000000G 055430                CMP     L$LNAM(R0),#<^RNW > ; CHECK LLC NAME
002012  001366                                BNE     10$               ; BR IF WRONG LLC
002014  032767  000000G 000000G               BIT     #XS.CTM,$OPRND    ; COUNTER TIMER ?
002022  001406                                BEQ     20$               ; BR IF NO
002024  016760  000000G 000000G               MOV     $CT,L$LCTM(R0)    ; UPDATE COUNTER TIMER
002032  052760  000000G 000000G               BIS     #CS.MOD,C.STS(R0) ; MARK TEMPLATE FOR UPDATE
002040  000207                        20$:    RETURN
                                      ;
                                      ; ERRORS
                                      ;
002042                                101$:   ERROR$  #ERRXSV           ; NO X25-SERVER DATABASE
002052  000207                                RETURN
```

```
                                                                  D 15
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29       PAGE 4
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME        REFERENCES

CALLR             #4-75      6-136
ERROR$            #4-64      6-184      9-326      9-328      9-330      10-368     11-396     13-490     13-492     13-494
                    14-566   14-569     15-585
PRINT             #4-59
```

```
                                                                  E 15
**FILE**ID**SETNOD
```

```
SETNOD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:22     PAGE 2

MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-76
ERROR$            #4-65        5-198
PRINT             #4-60
```

```
**FILE**ID**SETNS"
```

```
    472                                                .SBTTL  .MOPRI  - PARSE LIST INSERTION PRIORITY
    473                                        ;+
    474                                        ; ** .MOPRI - PARSE LIST INSERTION PRIORITY
    475                                        ;
    476                                        ;
    477                                        ; INPUTS         .TPARS VARAIBLES
    478                                        ;
    479                                        ; OUTPUT         LIST INSERTION PRIOITY STORED IN $PRI.
    480                                        ;-
    481
    482 001050                                 .MOPRI::
    483 001050  005767  000000G                        TST     .PNUMH          ; TEST FOR HIGH ORDER
    484 001054  001214                                 BNE     ILLNUM          ; IF NE, BAD NUMBER
    485 001056  105767  000001G                        TSTB    .PNUMB+1        ; CHECK HI BYTE
    486 001062  001211                                 BNE     ILLNUM          ; IF NE, BAD NUMBER
    487 001064  116767  000000G 000000G                MOVB    .PNUMB,$PRI     ; SAVE PRIORITY
    488 001072  000207                                 RETURN
```

```
                                                .SBTTL  .OBNAM - STORE THE OBJECT TASK NAME
                                        ;+
                                        ; *** - .OBNAM - STORE THE OBJECT TASK NAME
                                        ;
                                        ; INPUT:
                                        ;       .PSTPT = ADDRESS OF THE OBJECT TASK NAME
                                        ;       .PSTCN = NUMBER OF CHARACTERS IN THE TASK NAME
                                        ;
                                        ; OUTPUT:
                                        ;       $TNAM = THE OBJECT TASK NAME IN RAD50
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-
                                        
000036                                  .OBNAM::
000036  005067  000002G                         CLR     $TNAM+2         ; CLEAR THE 2ND HALF OF THE TASK NAME
000042  016700  000000G                         MOV     .PSTPT,R0       ; POINT AT THE TASK NAME
000046  010046                                  MOV     R0,-(SP)        ; COPY POINTER
000050  066716  000000G                         ADD     .PSTCN,(SP)     ; POINT TO LAST CHAR AFTER LAST
000054  117646  000000                          MOVB    @(SP),-(SP)     ; SAVE CHARACTER
000060  112776  000077  000002                  MOVB    #'?,@2(SP)      ; INSERT SCAN STOPPER
000066  010701                                  MOV     PC,R1           ; PERIODS ARE ACCEPTABLE
000070  004767  000000G                         CALL    $CAT5           ; CONVERT TASK NAME TO RAD50
000074  010167  000000G                         MOV     R1,$TNAM        ; STORE THE TASK NAME
000100  103414                                  BCS     10$             ; IF CS, END OF SCAN
000102  010701                                  MOV     PC,R1           ; ACCEPT PERIODS
000104  004767  000000G                         CALL    $CAT5           ; CONVERT 2ND HALF TO RAD50
000110  010167  000002G                         MOV     R1,$TNAM+2      ; STORE THE 2ND HALF OF THE TASK NAME
000114  022767  000006  000000G                 CMP     #6,.PSTCN       ; IS THE TASK NAME TOO BIG ?
000122  103003                                  BHIS    10$             ; IF HIS, NO .. EXIT
000124  062766  000002  000004                  ADD     #2,4(SP)        ; ELSE, REJECT
000132  112636                          10$:    MOVB    (SP)+,@(SP)+    ; RESTORE CHARACTER AND CLEAN STACK
000134  000207                                  RETURN
```

```
**FILE**ID**PRSREM
```

```
PPPPPPPP     RRRRRRRR       SSSSSSSS   RRRRRRRR     EEEEEEEEEE   MM      MM
PPPPPPPP     RRRRRRRR       SSSSSSSS   RRRRRRRR     EEEEEEEEEE   MM      MM
PP      PP   RR      RR   SS           RR      RR   EE           MMMM  MMMM
PP      PP   RR      RR   SS           RR      RR   EE           MMMM  MMMM
PP      PP   RR      RR   SS           RR      RR   EE           MM  MM  MM
PP      PP   RR      RR   SS           RR      RR   EE           MM  MM  MM
PPPPPPPP     RRRRRRRR       SSSSSS     RRRRRRRR     EEEEEEEE     MM      MM
PPPPPPPP     RRRRRRRR       SSSSSS     RRRRRRRR     EEEEEEEE     MM      MM
PP           RR  RR               SS   RR  RR       EE           MM      MM
PP           RR  RR               SS   RR  RR       EE           MM      MM
PP           RR    RR             SS   RR    RR     EE           MM      MM       ....
PP           RR    RR             SS   RR    RR     EE           MM      MM       ....
PP           RR      RR   SSSSSSSS     RR      RR   EEEEEEEEEE   MM      MM       ....
PP           RR      RR   SSSSSSSS     RR      RR   EEEEEEEEEE   MM      MM       ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                    E  4
PRSREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE             REFERENCES

CKFSP     000744 RG       #9-398
CS.REM  = ******   GX      5-133
C.ADD     000010          #4-73      5-135
C.LEN   = 000012          #4-74
C.NAM     000004          #4-72
C.STS   = ******   GX      5-133
D       = 000001          #7-259     7-265     7-265     7-265     7-265     7-265     7-265     7-265     7-265
                           7-265     7-265     7-265     7-265     7-268     7-268     7-268     7-268     7-268
                           7-268     7-268     7-268     7-268     7-268     7-268     7-268
E       = 000000          #7-258     7-265     7-268
ERRCID  = ******   GX      6-246
ERRILN  = ******   GX      6-219
ERRNOD  = ******   GX      5-142
ERRPNA  = ******   GX      5-139
F.DEV     000001          #4-61      9-417
F.LEN   = 000037          #4-62
H       = 177777          #7-260     7-265     7-265     7-265     7-265     7-265
HEX       000604 R         7-287    #7-330     8-362
ILLNUM    000310 R         6-210     6-212    #6-219
I$$AS   = ******           6-215
MOVSTR    001036 R         9-418    #9-432
MXFSP   = 000036          #4-56      4-61      9-399
REJ       000316 R        #6-221     6-247     7-279     8-359     8-363
REJ1      000730 R         7-284     7-288    #8-372     9-400     9-414
RN.NAM  = ******   GX      5-115
R$$EIS  = ******           6-215
R$$11D  = ******           6-215
SCR       000000 R        #4-80      7-273     7-292     8-352     8-367
STR50     000530 R         6-244    #7-304
STR50X    000540 R         7-295    #7-306     8-370
T1        000362 R        #7-265     7-274
T2        000405 R        #7-268     7-277
WC.CIR  = ******   GX      6-241
$CAT5   = ******   GX      6-180
$ERROR  = ******   GX     *5-139    *5-142    *6-219    *6-246
$NODAD  = ******   GX      5-126     5-135     6-158     6-165    *6-166
$NODNA  = ******   GX      5-117     5-119
$NTADD  = ******   GX      5-126
$NTNAM  = ******   GX      5-117     5-119
$OPRND  = ******   GX      5-115
$RNDCP  = ******   GX     *6-216
$RNDPA  = ******   GX     *6-194    *6-195
$RNHAR  = ******   GX      7-294
$RNHOS  = ******   GX      4-91     *6-158     6-164    *6-165
$RNSCI  = ******   GX      6-243
$RNSDV  = ******   GX     *6-181
$RNSPA  = ******   GX      8-369
$RNXXX  = ******   GX      4-93
$TMLST  = ******   GX      5-129
$WILD   = ******   GX      6-241
.CAT5E  = ******   GX      7-312
.PNUMB  = ******   GX      6-194     6-209
```

```
                                                                    F  4
PRSREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 2
```

```
                                                  .SBTTL  .SYSBN - SET NUMBER OF SDB'S
                                          ;+
                                          ; *** - .SYSBN - SET NUMBER OF SDB'S
                                          ;
                                          ; INPUT:
                                          ;       .PNUMB = NUMBER OF SDB'S
                                          ;       .PNUMH = NUMBER OF SDB'S (HIGH ORDER)
                                          ;
                                          ; OUTPUT:
                                          ;       $SDBNM = THE NUMBER OF SDB'S
                                          ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                          ;
                                          ;-
 000364                                   .SYSBN::
 000364  016767  000000G 000000G                  MOV     .PNUMB,$SDBNM   ; STORE THE NUMBER OF SDB'S
 000372  005767  000000G                          TST     .PNUMH          ; IS IT A WORD VALUE ?
 000376  001010                                   BNE     5$              ; IF NE, NO .. REJECT
 000400  022767  000003  000000G                  CMP     #3.,.PNUMB      ; THE NUMBER OF SDB'S MUST BE X WHERE:
 000406  003004                                   BGT     5$              ; 3. <= X <= 200.
 000410  022767  000310  000000G                  CMP     #200.,.PNUMB    ;         ...
 000416  002005                                   BGE     10$             ;         ...
 000420  062716  000002                   5$:     ADD     #2,(SP)         ; ELSE REJECT THE TRANSITION
 000424  012767  000000G 000000G                  MOV     #ERRILN,$ERROR  ; INVALID NUMBER
 000432  000207                           10$:    RETURN
```

```
                                                .SBTTL  CHKDTE - CHECK DTE ELIGIBILITY FOR DELETION
                                        ;+
                                        ; ** CHKDTE - CHECK DTE ELIGIBILITY FOR DELETION
                                        ;
                                        ; INPUTS:       R2 = ADDRESS OF DTE TEMPLATE
                                        ;
                                        ; OUTPUT:       CARRY CLEAR - DTE IS ELIGIBLE FOR DELETION
                                        ;               CARRY SET - DTE CANNOT BE DELETED (IT HAS DLM PVCS)
                                        ;               ERROR MESSAGE PRINTED
                                        ;
                                        ;-
001004                                  CHKDTE:
001004  004267  000000G                         JSR     R2,$SAVVR         ; SAVE VOLATILE REGISTERS
001010  010201                                  MOV     R2,R1             ; COPY TEMPLATE ADDRESS
001012  011101                          10$:    MOV     (R1),R1           ; NEXT TEMPLATE
001014  001437                                  BEQ     20$               ; BR IF END OF LIST
001016  126127  000000G 000000G                 CMPB    C.STS(R1),#CS.CHN ; CHANNEL DEF'N?
001024  001772                                  BEQ     10$               ; IF EQ, CONTINUE SCAN
001026  126127  000000G 000000G                 CMPB    C.STS(R1),#CS.CUG ; CUG DEF'N?
001034  001766                                  BEQ     10$               ; IF EQ, CONTINUE SCAN
001036  126127  000000G 000000G                 CMPB    C.STS(R1),#CS.PVC ; PVC DEF'N
001044  001023                                  BNE     20$               ; IF NE, END OF SCAN
001046  026127  000000G 015355                  CMP     P$VOWN(R1),#^RDLM ; DLM PVC ?
001054  001356                                  BNE     10$               ; IF NE, NO - STILL OK TO DELETE
001056  012700  000000G                         MOV     #$BUFF,R0         ; POINT TO MESSAGE BUFFER
001062  012701  000354'                         MOV     #ERRDLM,R1        ; COPY FORMATTING STRING
001066  112120                                  MOVB    (R1)+,(R0)+       ; MOVE STATUS BYTE TO OUTPUT BUFFER
001070  062702  000000G                         ADD     #D$TADR,R2        ; POINT TO RAD50 DTE ADDRESS
001074  004767  000000G                         CALL    $EDMSG            ; CONVERT RAD50 TO ASCII
001100                                          ERROR$  #$BUFF            ; ERROR - CAN'T DELETE DTE WITH DLM PVCS
001110  000261                                  SEC                       ; RETURN FAILURE
001112  000401                                  BR      30$
001114  000241                          20$:    CLC                       ; RETURN SUCCESS
001116  000207                          30$:    RETURN
```

```
**FILE**ID**PURREM
```

```
PPPPPPPP    UU      UU  RRRRRRRR    RRRRRRRR    EEEEEEEEEE  MM      MM
PPPPPPPP    UU      UU  RRRRRRRR    RRRRRRRR    EEEEEEEEEE  MM      MM
PP      PP  UU      UU  RR      RR  RR      RR  EE          MMMM  MMMM
PP      PP  UU      UU  RR      RR  RR      RR  EE          MMMM  MMMM
PP      PP  UU      UU  RR      RR  RR      RR  EE          MM  MM  MM
PP      PP  UU      UU  RR      RR  RR      RR  EE          MM  MM  MM
PPPPPPPP    UU      UU  RRRRRRRR    RRRRRRRR    EEEEEEEE    MM      MM
PPPPPPPP    UU      UU  RRRRRRRR    RRRRRRRR    EEEEEEEE    MM      MM
PP          UU      UU  RR  RR      RR  RR      EE          MM      MM
PP          UU      UU  RR  RR      RR  RR      EE          MM      MM
PP          UU      UU  RR    RR    RR    RR    EE          MM      MM      ....
PP          UU      UU  RR    RR    RR    RR    EE          MM      MM      ....
PP          UUUUUUUUUU  RR      RR  RR      RR  EEEEEEEEEE  MM      MM      ....
PP          UUUUUUUUUU  RR      RR  RR      RR  EEEEEEEEEE  MM      MM      ....
```

```
LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
                                                                 E  8
PURREM - PURGE REMOTE NODE      MACRO V05.03b Saturday 29-Jun-85 05:20  Page 11-1
Symbol table

CS.FIL= ****** GX       F.DEV   000005          RN.LOD= ****** GX       SF.HAD= 000004          S$PSS   000034
CS.REM= ****** GX       F.LEN1= 000043          RN.NAM= ****** GX       SF.HST= 000002          S.LENX= ****** GX
CS.SER= ****** GX       F.LNK   000000          RN.SCI= ****** GX       SF.PH3= 000400          WC.NOD= ****** GX
C.PRM = ****** GX       MXFSP = 000036          RN.SDV= ****** GX       SF.PSS= 000040          $BYTMN= ****** GX
C.STS = ****** GX       PRGFIL  000522R         RN.SLO= ****** GX       STBL    000460R         $BYTXT= ****** GX
DEAFIL  000614R         PRGSER  000432R         RN.SNV= ****** GX       SUCC    000000R         $CFERR= ****** GX
ERRNOD= ****** GX       PURDLL  000174R         RN.SPA= ****** GX       S$ADD   000006          $FILHD= ****** GX
FILPT   000004R         QREM    000710R         RN.TLO= ****** GX       S$CIR   000022          $NODAD= ****** GX
FTBL    000566R         REMLEN= 000012          R.LENX= ****** GX       S$DEV   000032          $OPRND= ****** GX
F$DIA   000012          RN.DCT= ****** GX       SERPT   000002R         S$DPA   000050          $PUREM  000006RG
F$DUM   000014          RN.DIA= ****** GX       SF.ADD= 000001          S$DPC   000054          $RLBF = ****** GX
F$LD    000010          RN.DMP= ****** GX       SF.CIR= 000010          S$FLG   000004          $TMLST= ****** GX
F$LEN = 000016          RN.DPA= ****** GX       SF.DEV= 000020          S$HAD   000012          $WILD = ****** GX
F$SEC   000004          RN.HAR= ****** GX       SF.DPA= 000100          S$HST   000010          $XXAVL= ****** GX
F$TER   000006          RN.HOS= ****** GX       SF.DPC= 000200          S$LEN = 000056          .CONER= ****** GX
F.CNT   000002

. ABS.  000056    000   (RW,I,GBL,ABS,OVR)
        000722    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8375 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.34
SY:PURREM.V2,[132,134]PURREM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PURREM
```

```
                                                                 F  8
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20     PAGE 1
```

```
SETBUF      CREATED BY  MACRO  ON 29-JUN-85 AT 05:20     PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME        REFERENCES

CHADF$            #4-89        4-91
EPRINT            #4-76
ERROR$            #4-65        5-199        5-223
PRINT             #4-60
```

```
**FILE**ID**SETCNT
```

```
				.TITLE	SETCUG - SET CLOSED USER GROUP PARAMETERS
				.IDENT	/V05.00/
				.ENABL	LC

			;
			; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
			; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
			;
			;
			; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
			; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
			; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
			; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
			; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
			; TRANSFERRED.
			;
			; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
			; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
			; CORPORATION.
			;
			; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
			; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
			;
			;
			; MODULE DESCRIPTION:
			;
			; CFE - SET CLOSED USER GROUP PARAMETERS
			;
			;
			; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
			;
			; IDENT HISTORY:
			;
			; 1.00	26-OCT-81
			;	DECNET-11M/S V3.1
			;	DECNET-11M-PLUS V1.1
			;
			; 3.00	16-APR-82
			;	DECNET-11M V3.1
			;	DECNET-11M-PLUS V1.1
			;
			; 4.00	07-NOV-83
			;	DECNET-11M V4.0
			;	DECNET-11M-PLUS V2.0
			;
			; 5.00	22-JUL-85
			;	DECnet-11M/S V4.2
			;	DECnet-11M-Plus V3.0
			;	DECnet-Micro/RSX V1.0
			;
			;

```

```
     82                                                          .SBTTL  LOCAL DATA
     83                                                          .NLIST  BEX
     84                                                  ;
     85                                                  ; LOCAL DATA
     86                                                  ;
     87 000000     000G      000G      000G  PRDLST:  .BYTE   CS.PSN, CS.X3P, CS.X2P, CS.x29, CS.DTE
     88 000005     000G      000G      000G           .BYTE   CS.CHN, CS.PVC, CS.SVC, CS.CUG, 0
     89
     90        000000                                      EX$WAR = 0
     91        000002                                      EX$ERR = 2
     92 000012     002       103       106   ERRPMI:  .ASCIZ  <EX$ERR>/CFE -- Missing parameter: LINE/
     93 000052     000       103       106   ERRLIN:  .ASCII  <EX$WAR>/CFE -- DTE %5R not enabled,/
     94 000106     040       157       164            .ASCIZ           / other DTE using line/
     95                                                .EVEN
     96                                                .LIST BEX
     97
```

Table of contents

```
     4-   54     MACRO CALLS
     5-   95     LOCAL DATA
     6-  116     LOCAL SYMBOL DEFINITIONS
     7-  144     $SELOG - SET LOGGING PARAMETERS
     8-  260     FINEVT - FIND NEXT EVENT IN LIST
     9-  285     PROEVT - PROCESS EVENT MASK
```

```
SETLOG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME       REFERENCES

EPRINT           #4-81
ERROR$           #4-70       7-175       7-215       7-241
FLTDF$           #4-91       4-93
PRINT            #4-65
```

```
**FILE**ID**SETMOD
```

```
                                        .SBTTL  X29CHA - UPDATE X29-SERVER CHARACTERISTICS
                                ;+
                                ; ** X29CHA - UPDATE X29-SERVER CHARACTERISTICS
                                ;
                                ; INPUTS        NONE.
                                ;
                                ; OUTPUT        THE X29-SERVER CHARACTERISTICS ARE UPDATED.
                                ;-

002054                          X29CHA:
002054  012700  000000G                 MOV     #$TMLST,R0        ; POINT TO LISTHEAD
002060  011000                  10$:    MOV     (R0),R0           ; NEXT
002062  001423                          BEQ     40$               ; IF EQ, NO X29 SUPPORT
002064  122760  000000G 000000G         CMPB    #CS.X29,C.STS(R0) ; CHECK FOR X29 DATABASE
002072  001372                          BNE     10$               ; KEEP ON LOOKING
002074  032767  000000G 000000G         BIT     #XS.CTM,$OPRND    ; COUNTER TIMER ?
002102  001403                          BEQ     20$               ; BR IF NO
002104  016760  000000G 000000G         MOV     $CT,X$9CT(R0)     ; UPDATE COUNTER TIMER
002112  032767  000000G 000000G 20$:    BIT     #XS.MXC,$OPRND    ; MAX CIRCUITS?
002120  001403                          BEQ     30$               ; BR IF NO
002122  016760  000000G 000000G         MOV     $MCI,X$9MXC(R0)   ; UPDATE
002130  000207                  30$:    RETURN

                                ; ERRORS
                                ;
002132                          40$:    ERROR$  #ERRX29           ; NO X29 SUPPORT
002142  000207                          RETURN

```

```
**FILE**ID**SETNOD




  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  NN      NN    OOOOOO    DDDDDDDD
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  NN      NN    OOOOOO    DDDDDDDD
SS          EE              TT      NN      NN  OO      OO  DD      DD
SS          EE              TT      NN      NN  OO      OO  DD      DD
SS          EE              TT      NNNN    NN  OO      OO  DD      DD
SS          EE              TT      NNNN    NN  OO      OO  DD      DD
  SSSSSS    EEEEEEEE        TT      NN  NN  NN  OO      OO  DD      DD
  SSSSSS    EEEEEEEE        TT      NN  NN  NN  OO      OO  DD      DD
        SS  EE              TT      NN    NNNN  OO      OO  DD      DD
        SS  EE              TT      NN    NNNN  OO      OO  DD      DD
        SS  EE              TT      NN      NN  OO      OO  DD      DD      ....
        SS  EE              TT      NN      NN  OO      OO  DD      DD      ....
SSSSSSSS    EEEEEEEEEE      TT      NN      NN    OOOOOO    DDDDDDDD        ....
SSSSSSSS    EEEEEEEEEE      TT      NN      NN    OOOOOO    DDDDDDDD        ....




LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
**FILE**ID**SETNSP
```

```
  SSSSSSSS    EEEEEEEEEE    TTTTTTTTTT    NN      NN      SSSSSSSS    PPPPPPPP
  SSSSSSSS    EEEEEEEEEE    TTTTTTTTTT    NN      NN      SSSSSSSS    PPPPPPPP
SS            EE                TT        NN      NN    SS            PP      PP
SS            EE                TT        NN      NN    SS            PP      PP
SS            EE                TT        NNNN    NN    SS            PP      PP
SS            EE                TT        NNNN    NN    SS            PP      PP
  SSSSSS      EEEEEEEE          TT        NN  NN  NN      SSSSSS      PPPPPPPP
  SSSSSS      EEEEEEEE          TT        NN  NN  NN      SSSSSS      PPPPPPPP
        SS    EE                TT        NN    NNNN            SS    PP
        SS    EE                TT        NN    NNNN            SS    PP
        SS    EE                TT        NN      NN            SS    PP              ....
        SS    EE                TT        NN      NN            SS    PP              ....
SSSSSSSS      EEEEEEEEEE        TT        NN      NN    SSSSSSSS      PP              ....
SSSSSSSS      EEEEEEEEEE        TT        NN      NN    SSSSSSSS      PP              ....


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

```
                                                .SBTTL  .MODS1  - PARSE DESTINATION SUBADDRESS RANGE
                                                .SBTTL  .MODS2  - PARSE DESTINATION SUBADDRESS RANGE
                                        ;+
                                        ; ** .MODS1 - PARSE DESTINATION SUBADDRESS RANGE
                                        ; ** .MODS2 - PARSE DESTINATION SUBADDRESS RANGE
                                        ;
                                        ; INPUTS        .TPARS VARAIBLES
                                        ;
                                        ; OUTPUT        THE DESTINATION SUBADDRESS IS SAVED IN
                                        ;               THE PARSED DATA AREA
                                        ;
                                        ;-
                                                .ENABL  LSB
001074                                  .MODS1::
001074  012700  000000G                         MOV     #$SUB1,R0       ; TARGET ADDRESS
001100  012767  177777  000000G                 MOV     #-1,$SUB2       ; NO SECOND ADDRESS, YET
001106  000402                                  BR      10$
001110                                  .MODS2::
001110  012700  000000G                         MOV     #$SUB2,R0       ; TARGET ADDRESS
001114  005767  000000G                 10$:    TST     .PNUMH          ; CHECK HIGH WORD
001120  001006                                  BNE     ILLNU2          ; BR IF BAD NUMBER
001122  016710  000000G                         MOV     .PNUMB,(R0)     ; SAVE NUMBER
001126  021027  023417                          CMP     (R0),#9999.     ; CHECK HIGH BOUND
001132  101001                                  BHI     ILLNU2          ; BR IF OUT OF RANGE
001134  000207                                  RETURN

                                                .DSABL  LSB

001136  012767  000000G 000000G ILLNU2: MOV     #ERRILN,$ERROR
001144  062716  000002          REJEC2: ADD     #2,(SP)
001150  000207                                  RETURN

```

```
                                          .SBTTL  .OBCOP - STORE THE NUMBER OF OBJECT TASK COPIES
                                  ;+
                                  ; *** - .OBCOP - STORE THE NUMBER OF OBJECT TASK COPIES
                                  ;
                                  ; INPUT:
                                  ;       .PNUMB = THE NUMBER OF OBJECT TASK COPIES
                                  ;       .PNUMH = THE NUMBER OF OBJECT TASK COPIES (HIGH ORDER)
                                  ;
                                  ; OUTPUT:
                                  ;       $COPES = THE NUMBER OF OBJECT TASK COPIES
                                  ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                  ;
                                  ;-

000136                            .OBCOP::
000136  016767  000000G 000000G           MOV     .PNUMB,$COPES   ; STORE THE NUMBER OF COPIES
000144  001413                            BEQ     10$             ; IF EQ, BAD NUMBER OF COPIES
000146  005767  000000G                   TST     .PNUMH          ; IS THIS A WORD VALUE ?
000152  001010                            BNE     10$             ; IF NE, NO .. REJECT
000154  022767  000100  000000G           CMP     #100,.PNUMB     ; IS THE NUMBER OF COPIES <= 100
000162  103404                            BLO     10$             ; IF LO, NO .. REJECT
000164  001010                            BNE     20$             ; IF NE, IT IS STRICTLY < 100
000166  005067  000000G                   CLR     $COPES          ; ELSE, ZERO THE NUMBER OF COPIES
000172  000405                            BR      20$             ; AND LEAVE
000174  062716  000002            10$:    ADD     #2,(SP)         ; REJECT THE TRANSITION
000200  012767  000000G 000000G           MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000206  000207                    20$:    RETURN


        000001                            .END
```

Table of contents

```
     4-    51     DEFINITIONS AND DATA AREA
     4-    82     .RNIN2 - INITIALIZE REMOTE NODE OPTIONS
     5-    97     .RNCHK - CHECK REMOTE NODE OPTIONS
     6-   149     .RNHO1  - PARSE REMOTE NODE HOST
     6-   150     .RNHOS  - PARSE REMOTE NODE HOST
     6-   170     .RNSDV - PARSE SERVICE DEVICE
     6-   185     .RNDPA - PARSE DUMP ADDRESS
     6-   199     .RNDCT - PARSE DUMP COUNT
     6-   227     .RNSCI - STORE SERVICE CIRCUIT
     7-   249     .RNHAR - STORE HARDWARE ADDRESS
     8-   342     .RNSPA - STORE SERVICE PASSWORD
     9-   376     FILE PARSING ACTION ROUTINES
```

```
                                                                 F  4
PRSREM      CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 2

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE               REFERENCES

.PNUMH   = ******   GX      6-195      6-211
.PSTCN   = ******   GX      6-239      7-278      7-305      8-356      9-399      9-403      9-433
.PSTPT   = ******   GX      6-179      7-272      7-304      8-360      9-402      9-434
.RNCHK     000036 RG       #5-111
.RNCLF     000736 RG       #9-386
.RNDCT     000246 RG       #6-208
.RNDPA     000230 RG       #6-193
.RNFSP   = ******   GX     *9-387      9-416
.RNHAR     000424 RG       #7-271
.RNHOS     000172 RG       #6-163
.RNHO1     000156 RG       #6-157
.RNHO2     000166 RG       #6-161
.RNIN2     000020 RG       #4-90
.RNSCI     000324 RG       #6-238
.RNSDV     000212 RG       #6-178
.RNSPA     000642 RG       #8-351
```

```
                                                                 G  4
PRSREM      CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 3
```

```
                                                .SBTTL  .SYTHR - SET MINIMUM NUMBER OF RDB'S FOR POOL
                                        ;+
                                        ; *** - .SYTHR - SET MINIMUM NUMBER OF RDB'S FOR POOL
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = MINIMUM NUMBER OF RDB'S FOR POOL
                                        ;       .PNUMH = MINIMUM NUMBER OF RDB'S FOR POOL (HIGH ORDER)
                                        ;
                                        ; OUTPUT:
                                        ;       $THRSH = THE MINIMUM NUMBER OF RDB'S FOR POOL
                                        ;       THE TRANSITION IS ACCEPTED/REJECTED.
                                        ;
                                        ;-

000434                                  .SYTHR::
000434  016767  000000G 000000G                 MOV     .PNUMB,$THRSH   ; STORE THE MINIMUM NUMBER OF RDB'S FOR POOL
000442  005767  000000G                         TST     .PNUMH          ; IS IT A WORD VALUE ?
000446  001012                                  BNE     5$              ; IF NE, NO .. REJECT
000450  022767  000001  000000G                 CMP     #1.,.PNUMB      ; THE MIN. # OF RDB'S MUST BE X WHERE:
000456  003006                                  BGT     5$              ; 1. <= X <= (NUMBER OF RDB'S - 1.)
000460  016700  000000G                         MOV     $RDBNM,R0       ;       ...
000464  005300                                  DEC     R0              ;       ...
000466  020067  000000G                         CMP     R0,.PNUMB       ;       ...
000472  002005                                  BGE     10$             ;       ...
000474  062716  000002                  5$:     ADD     #2,(SP)         ; ELSE REJECT THE TRANSITION
000500  012767  000000G 000000G                 MOV     #ERRILN,$ERROR  ; INVALID NUMBER
000506  000207                          10$:    RETURN


        000001                                  .END
```

```
                                                        .SBTTL  PURCUG - PURGE CLOSED USER GROUP
                                                ;+
                                                ; ** PURDTE - PURCUG - PURGE CLOSED USER GROUP
                                                ;
                                                ; INPUTS        $OPRND = REQUEST TYPE
                                                ;
                                                ; OUTPUT        THE REQUESTED CUG DESCRIPTOR(S) ARE PURGED.
                                                ;-

001120                                          PURCUG:
001120  005004                                          CLR     R4                      ; NONE FOUND YET
001122  012702  000000G                                 MOV     #$TMLST,R2              ; POINT TO LISTHEAD
001126  010205                                  10$:    MOV     R2,R5                   ; SAVE PREVIOUS ADDRESS
001130  011202                                          MOV     (R2),R2                 ; NEXT ADDRESS
001132  001431                                          BEQ     30$                     ; BR IF END OF LIST
001134  122762  000000G 000000G                         CMPB    #CS.CUG,C.STS(R2)       ; CHECK FOR DTE BLOCK
001142  001371                                          BNE     10$                     ; BR UNLESS DTE
001144  032767  000000G 000000G                         BIT     #XP.KNG,$OPRND          ; CHECK FOR KNOWN DTES
001152  001010                                          BNE     20$                     ; BR IF WILDCARD
001154  026762  000000G 000000G                         CMP     $CUG,C$UNAM(R2)         ; CHECK FIRST PART OF NAME
001162  001361                                          BNE     10$                     ; BR IF WRONG CUG
001164  026762  000002G 000002G                         CMP     $CUG+2,C$UNAM+2(R2) ; COMPARE SECOND WORD
001172  001355                                          BNE     10$                     ; IF NE, KEEP LOOKING
001174  010604                                  20$:    MOV     SP,R4                   ; INDICATE WILDCARDS OR FOUND
001176  012700  000000G                                 MOV     #C$LENX,R0              ; LENGTH OF CUG IN POOL..
001202  012701  000000G                                 MOV     #C$ULEN,R1              ; LENGTH OF A CUG TEMPLATE
001206  004767  000652                                  CALL    DEALC                   ; DEALLOCATE IT
001212  010502                                          MOV     R5,R2                   ; RESUME SEARCH CORRECTLY
001214  000744                                          BR      10$                     ; CONTINUE
001216  032767  000000G 000000G                 30$:    BIT     #XP.KNG,$OPRND          ; WILDCARD OPERATION ?
001224  001002                                          BNE     40$                     ; IF NE, YES
001226  005704                                          TST     R4                      ; ANYTHING FOUND
001230  001401                                          BEQ     101$                    ; IF EQ, NOPE
001232  000207                                  40$:    RETURN
                                                ;
                                                ; ERRORS
                                                ;
001234                                          101$:   ERROR$  #ERRNSG                 ; NO SUCH GROUP
001244  000207                                          RETURN

```

Table of contents

```
    4-   45     MACRO DEFINITIONS
    5-   70     LOCAL DATA
    6-  144     PURREM - PURGE REMOTE
    7-  200     PURDLL - PURGE DLL INFO
    8-  268     PRGSER - PURGE SERVICE BLOCK PARAMETERS
    9-  302     PRGFIL - PURGE SERVICE FILE BLOCKS
   10-  338     DEAFIL - DEALLOCATE A FILE BLOCK
   11-  376     QREM - REMOVE ENTRY FROM QUEUE
```

```
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE              REFERENCES

CS.FIL  = ******  GX        7-226
CS.REM  = ******  GX        6-166
CS.SER  = ******  GX        7-219
C.PRM   = ******  GX        6-170
C.STS   = ******  GX        6-166       7-219       7-226
DEAFIL    000614 R          9-323      #10-353
ERRNOD  = ******  GX        6-197
FILPT     000004 R         #5-142      *7-225       7-240       7-257       9-319
FTBL      000566 R          9-314      #9-329
F$DIA     000012           #5-118       9-333
F$DUM     000014           #5-119       9-334
F$LD      000010           #5-117       9-332
F$LEN   = 000016           #5-120       7-259
F$SEC     000004           #5-115       9-330
F$TER     000006           #5-116       9-331
F.CNT     000002           #5-131      *10-353
F.DEV     000005           #5-133       10-356
F.LEN1  = 000043           #5-134       10-369
F.LNK     000000           #5-130
MXFSP   = 000036           #5-126       5-133
PRGFIL    000522 R          7-232       7-256      #9-313
PRGSER    000432 R          7-231      #8-280
PURDLL    000174 R          6-185      #7-210
QREM      000710 R          6-179       7-249       7-258      #11-391
REMLEN  = 000012           #5-139       6-180
RN.DCT  = ******  GX        8-297
RN.DIA  = ******  GX        7-255       9-333
    DMP = ******  GX        7-255       9-334
    DPA = ******  GX        8-296
   .HAR = ******  GX        8-292
RN.HOS  = ******  GX        8-291
RN.LOD  = ******  GX        7-255       9-332
RN.NAM  = ******  GX        6-177       7-229
RN.SCI  = ******  GX        8-293
RN.SDV  = ******  GX        8-294
RN.SLO  = ******  GX        7-255       9-330
RN.SNV  = ******  GX        8-298
RN.SPA  = ******  GX        8-295
RN.TLO  = ******  GX        7-255       9-331
R.LENX  = ******  GX        6-183       6-184
SERPT     000002 R         #5-141      *7-223       7-237
SF.ADD  = 000001           #5-98        7-238
SF.CIR  = 000010           #5-101       8-293
SF.DEV  = 000020           #5-102       8-294
SF.DPA  = 000100           #5-104       8-296
SF.DPC  = 000200           #5-105       8-297
SF.HAD  = 000004           #5-100       8-292
SF.HST  = 000002           #5-99        8-291
SF.PH3  = 000400           #5-106       8-298
SF.PSS  = 000040           #5-103       8-295
STBL      000460 R          8-281      #8-290
SUCC      000000 R         #5-140      *6-159      *6-176       6-189
```

```
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 2
```

F 9

```
**FILE**ID**SETCNT




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     CCCCCCCC   NN      NN   TTTTTTTTTT
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     CCCCCCCC   NN      NN   TTTTTTTTTT
SS           EE               TT       CC           NN      NN       TT
SS           EE               TT       CC           NN      NN       TT
SS           EE               TT       CC           NNNN    NN       TT
SS           EE               TT       CC           NNNN    NN       TT
  SSSSSS     EEEEEEEE         TT       CC           NN  NN  NN       TT
  SSSSSS     EEEEEEEE         TT       CC           NN  NN  NN       TT
        SS   EE               TT       CC           NN    NNNN       TT
        SS   EE               TT       CC           NN    NNNN       TT
        SS   EE               TT       CC           NN      NN       TT           ....
        SS   EE               TT       CC           NN      NN       TT           ....
SSSSSSSS     EEEEEEEEEE       TT         CCCCCCCC   NN      NN       TT           ....
SSSSSSSS     EEEEEEEEEE       TT         CCCCCCCC   NN      NN       TT           ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

G 9

SETCNT - SET CONTROLLER OPTIONS MACRO V05.03b Saturday 29-Jun-85 05:21

```
;****
; LOCAL MACROS
;****

        .MACRO  PRINT   TEXT
        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
        CALL    $PRINT          ; PRINT MESSAGE
        .ENDM   PRINT

        .MACRO  ERROR$  TEXT
;
; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
; MESSAGE STRING.
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
        .ENDC
        CALL    $CFERR                  ; PRINT ERROR MESSAGE
        .ENDM   ERROR$

        .MACRO  EPRINT  TEXT
;
; PRINT TEXT ON ERROR LUN
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
        .ENDC
        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
        .ENDM   EPRINT


```

```
                                                .SBTTL  $SEDTE - SET LOCAL DTE PARAMETERS
                                        ;+
                                        ; *** - $SEDTE - SET LOCAL DTE PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       THE DTE$DF TEMPLATE IS UPDATED
                                        ;
                                        ;-

000134                                  $SEDTE::
                                        ;
                                        ; LOOK FOR EXISTING DTE TEMPLATE
                                        ;
000134  012705  000000G                         MOV     #$TMLST,R5      ; GET START OF TEMPLATE LIST
000140  011505                          10$:    MOV     (R5),R5         ; NEXT TEMPLATE
000142  001424                                  BEQ     20$             ; IF EQ, NO MORE
000144  122765  000000G 000000G                 CMPB    #CS.DTE,C.STS(R5) ; IS THIS A DTE$DF TEMPLATE ?
000152  001372                                  BNE     10$             ; IF NE, NO - KEEP LOOKING
000154  032767  000000G 000000G                 BIT     #XP.KND,$OPRND  ; IS THIS KNOWN DTES ?
000162  001074                                  BNE     40$             ; IF NE, YES
000164  012700  000000G                         MOV     #$DTE,R0        ; POINT TO PARSED DTE STRING
000170  010502                                  MOV     R5,R2           ; COPY TEMPLATE POINTER
000172  062702  000000G                         ADD     #D$TADR,R2      ; POINT TO DTE STRING
000176  012701  000005                          MOV     #5,R1           ; LENGTH OF STRING IN WORDS
000202  022022                          15$:    CMP     (R0)+,(R2)+     ; COMPARE 3 CHARACTERS
000204  001355                                  BNE     10$             ; IF NE, NEXT TEMPLATE
000206  005301                                  DEC     R1              ; COUNTDOWN
000210  003374                                  BGT     15$             ; UNTIL STRINGS COMPARED
000212  000442                                  BR      30$             ; BR IF DTE ALREADY IN CONFIGURATION
                                        ;
                                        ;  DTE TEMPLATE NOT IN SYSTEM
                                        ;
000214  032767  000000G 000000G         20$:    BIT     #XP.KND,$OPRND  ; KNOWN DTES ?
000222  001133                                  BNE     99$             ; IF NE, YES - ALL DONE
000224  032767  000000G 000000G                 BIT     #XP.LIN,$SWICH  ; LINE ID IS REQUIRED FOR NEW TEMPLATE
000232  001535                                  BEQ     102$            ; IF EQ, PARAMETER MISSING
000234  012700  000000G                         MOV     #$XXAVL,R0      ; POINT AT END OF TASK LISTHEAD
000240  012701  000000G                         MOV     #D$TLEN,R1      ; GET LENGTH OF A DTE TEMPLATE
000244  004767  000000G                         CALL    $RQBF           ; TRY TO ALLOCATE A TEMPLATE
000250  103521                                  BCS     101$            ; IF CS, ERROR
000252  062767  000200G 000000G                 ADD     #L$LENX+<DEFHSH*4>,$BYTMN ; UPDATE BYTE MINIMUM
000260  062767  000200G 000000G                 ADD     #L$LENX+<DEFHSH*4>,$BYTXT ; UPDATE BYTE ALLOCATION
000266  012760  000000G 000000G                 MOV     #CS.DTE,C.STS(R0) ; INDICATE ITS A DTE TEMPLATE
000274  005060  000000G                         CLR     D$TCT(R0)       ; INITIALIZE COUNTER TIMER
000300  005060  000000G                         CLR     D$TSTA(R0)      ; INITIALLY OFF
000304  012760  000040  000000G                 MOV     #DEFHSH,D$THSH(R0); HASH TABLE SIZE
000312  010005                                  MOV     R0,R5           ; COPY POINTER
000314  004767  000222                          CALL    INSDTE          ; INSERT DTE TEMPLATE IN THE RIGHT PLACE
000320  032767  000000G 000000G         30$:    BIT     #XP.DTE,$OPRND  ; ID STRING SPECIFIED ?
000326  001412                                  BEQ     40$             ; IF EQ, NO
000330  012700  000000G                         MOV     #$DTE,R0        ; POINT TO PARSED DTE ADDRESS
000334  010501                                  MOV     R5,R1           ; COPY POINTER
000336  062701  000000G                         ADD     #D$TADR,R1      ; POINT TO DTE ADDRESS AREA
```

```
        .TITLE  SETLOG  - SET LOGGING PARAMETERS
        .IDENT  /V05.00/
        .NLIST  BEX

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SET EVENT PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

F 13

```
**FILE**ID**SETMOD




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   MM      MM     OOOOOO     DDDDDDDD
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   MM      MM     OOOOOO     DDDDDDDD
SS           EE               TT       MMMM  MMMM   OO      OO   DD      DD
SS           EE               TT       MMMM  MMMM   OO      OO   DD      DD
SS           EE               TT       MM  MM  MM   OO      OO   DD      DD
SS           EE               TT       MM  MM  MM   OO      OO   DD      DD
  SSSSSS     EEEEEEEE         TT       MM      MM   OO      OO   DD      DD
  SSSSSS     EEEEEEEE         TT       MM      MM   OO      OO   DD      DD
        SS   EE               TT       MM      MM   OO      OO   DD      DD
        SS   EE               TT       MM      MM   OO      OO   DD      DD
        SS   EE               TT       MM      MM   OO      OO   DD      DD         ....
        SS   EE               TT       MM      MM   OO      OO   DD      DD         ....
SSSSSSSS     EEEEEEEEEE       TT       MM      MM     OOOOOO     DDDDDDDD           ....
SSSSSSSS     EEEEEEEEEE       TT       MM      MM     OOOOOO     DDDDDDDD           ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

G 13

```
SETMOD - DEFINE MODULE  MACRO V05.03b Monday 01-Jul-85 17:29
```

```
                                                 .SBTTL  CHKDST  - CHECK FOR DESTINATION TEMPLATE
                                         ;+
                                         ; ** CHKDST - CHECK FOR DESTINATION TEMPLATE
                                         ;
                                         ; INPUTS         R2 = TEMPLATE ADDRESS
                                         ;                R3 = LINK ADDRESS
                                         ;
                                         ; OUTPUT         R3 = NEW LINK ADDRESS
                                         ;                CARRY CLEAR IF TEMPLATE IS DST$DF
                                         ;
                                         ; ** THIS CODE DEPENDS UPON THE ORDERING OF TEMPLATE TYPE CODES
                                         ;-

 002144  010201                          CHKDST: MOV     R2,R1                   ; COPY POINTER
 002146  005721                                  TST     (R1)+                   ; POINT TO C.STS
 002150  005703                                  TST     R3                      ; IS THERE ANY LINK ADDRESS?
 002152  001001                                  BNE     10$                     ; IF NE, YES
 002154  010503                                  MOV     R5,R3                   ; ELSE DEFAULT TO PREVIOUS
 002156  122711  000000G                 10$:    CMPB    #CS.DST,(R1)            ; CHECK FOR DESTINATION
 002162  001417                                  BEQ     40$                     ; IF EQ, GOT ONE
 002164  122711  000000G                         CMPB    #CS.DSC,(R1)            ; GOOD PLACE TO LINK NEW BLOCK?
 002170  001412                                  BEQ     20$                     ; IF EQ, YES
 002172  121127  000000G                         CMPB    (R1),#CS.PSN            ; BEFORE PSI TEMPLATES ?
 002176  002410                                  BLT     30$                     ; IF LT, DON'T LINK HERE
 002200  121127  000000G                         CMPB    (R1),#CS.DSC            ; AFTER ACCEPTABLE LINK LOCATIONS?
 002204  101005                                  BHI     30$                     ; IF HI, DON'T LINK HERE
 002206  122763  000000G 000000G                 CMPB    #CS.DSC,C.STS(R3)       ; IS OLD LINK ADDRESS BETTER ?
 002214  001401                                  BEQ     30$                     ; IF EQ, YES
 002216  010203                          20$:    MOV     R2,R3                   ; UPDATE LINK ADDRESS
 002220  005727                          30$:    TST     (PC)+                   ; RETURN NOT FOUND
 002222  000261                          40$:    SEC                             ; RETURN FOUND
 002224  000207                                  RETURN

```

Table of contents

```
     5-  136    $SENOD - SET NODE PARAMETERS
     6-  201    CHKADD - CHECK EXECUTOR NODE
     7-  231    UPDREM - UPDATE AREA OF REMOTE DEFINITIONS
```

Table of contents

5- 105 $SENSP - SET NSP PARAMETERS

```
                                                       .SBTTL  .MOOBN  - PARSE OBJECT NUMBER
                                             ;+
                                             ; ** .MOOBN        - PARSE OBJECT NUMBER
                                             ;
                                             ; INPUTS           .TPARS VARAIBLES
                                             ;
                                             ; OUTPUT           OBJECT NUMBER STORED IN $OBJ
                                             ;-

 001152                                      .MOOBN::
 001152  005767  000000G                               TST     .PNUMH          ; HIGH ORDER WORD ?
 001156  001367                                        BNE     ILLNU2          ; BR IF TOO HIGH
 001160  105767  000001G                               TSTB    .PNUMB+1        ; HIGH ORDER BYTE ?
 001164  001364                                        BNE     ILLNU2          ; BR IF TOO HIGH
 001166  116767  000000G 000000G                       MOVB    .PNUMB,$OBJ     ; SAVE OBJECT NUMBER
 001174  000207                                        RETURN
```

```
                                                                            G 2
PRSOBJ - PARSE OBJECTS ACTION R MACRO V05.03b Saturday 29-Jun-85 05:19  Page 7-1
Symbol table

ERRILN= ****** GX       $ERROR= ****** GX       .OBCOP  000136RG        .PNUMB= ****** GX       .PSTCN= ****** GX
$CAT5 = ****** GX       $TNAM = ****** GX       .OBNAM  000036RG        .PNUMH= ****** GX       .PSTPT= ****** GX
$COPES= ****** GX       $TNUM = ****** GX       .OBNUM  000000RG

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000210    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 117 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:04.35
SY:PRSOBJ.V2,[132,134]PRSOBJ/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSOBJ
```

```
                                                                            H 2
```

```
                        .TITLE  PRSREM - PARSE ACTION ROUTINES
                        .IDENT  /V05.00/
                        .ENABL  LC
                ;
                ; COPYRIGHT (C) 1983, 1984, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - PARSE ACTION ROUTINES
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  01-AUG-83
                ;       CREATED FOR NEW REMOTE NODE PARAMETERS.
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;


                        .MCALL  SAVRG,RESRG,ASR$

```

```
                                                                        G  4
PRSREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 3
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME      REFERENCES

ASR$            #3-48       6-215
RESRG           #3-48       7-320
SAVRG           #3-48       7-306
```

```
                                                                        H  4
```

```
                                                                                   G 5
PRSSYS - PARSE SYSTEM ACTION RO MACRO V05.03b Saturday 29-Jun-85 05:20  Page 12-1
Symbol table

CS.ROU= ****** GX        $ERROR= ****** GX        $SDBNM= ****** GX        .PSTCN= ****** GX        .SYPOL   000052RG
C.STS = ****** GX        $PARNA= ****** GX        $THRSH= ****** GX        .PSTPT= ****** GX        .SYRBN   000226RG
ERRILN= ****** GX        $POLNA= ****** GX        $TMLST= ****** GX        .SYBYT   000124RG        .SYRBS   000266RG
$BYT  = ****** GX        $RDBNM= ****** GX        .PNUMB= ****** GX        .SYCBN   000156RG        .SYSBN   000364RG
$CAT5 = ****** GX        $RDBSZ= ****** GX        .PNUMH= ****** GX        .SYPAR   000000RG        .SYTHR   000434RG
$CCBNM= ****** GX

. ABS.  000000     000   (RW,I,GBL,ABS,OVR)
        000510     001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 117 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.19
SY:PRSSYS.V2,[132,134]PRSSYS/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PRSSYS
```

```
                                            .SBTTL  $PUXSR- PURGE MODULE X25-SERVER
                                            .SBTTL  $PUX9S- PURGE MODULE X29-SERVER
                                    ;+
                                    ; *** - $PUXSR- PURGE MODULE X25-SERVER
                                    ; *** - $PUX9S- PURGE MODULE X29-SERVER
                                    ;
                                    ; INPUT:
                                    ;       $OPRND = REQUEST CODE
                                    ;       $DST   = PARSED DESTINATION NAME
                                    ;
                                    ; OUTPUT:
                                    ;       THE REQUESTED DESTINATION DESCRIPTOR AND THE
                                    ;       COMPANION DESTINATION ADDRESS AND DESTINATION
                                    ;       CALL DESCRIPTORS ARE PURGED.
                                    ;-

                                            .ENABL  LSB

 001246                             $PUXSV::
 001246  012704  115443                     MOV     #<^RX25>,R4     ; ONLY X25 DESTINATIONS
 001252  012702  001572'                    MOV     #XSVCTM,R2      ; ACTION ROUTINE IF COUNTER TIMER
 001256  000404                             BR      10$             ; BR IF NO

 001260                             $PUXS9::
 001260  012704  115447                     MOV     #<^RX29>,R4     ; ONLY X29 DESTINATIONS
 001264  012702  001650'                    MOV     #X29CTM,R2      ; IN CASE PURGE COUNTER TIMER
 001270  004767  000546             10$:    CALL    TSTPSI          ; MAKE SURE THIS IS A PSI SYSTEM
 001274  005003                             CLR     R3              ; NONE FOUND YET
 001276  032767  000000G 000000G            BIT     #XS.CTM,$OPRND  ; IS THIS PURGE COUNTER TIMER
 001304  001401                             BEQ     20$             ; IF EQ, NO
 001306                                     CALLR   (R2)            ; ELSE DISPATCH TO ROUTINE
 001310  012702  000000G            20$:    MOV     #$TMLST,R2      ; POINT TO LISTHEAD
 001314  010205                     30$:    MOV     R2,R5           ; SAVE PREVIOUS ADDRESS
 001316  011202                             MOV     (R2),R2         ; TRAVERSE LIST
 001320  001502                             BEQ     100$            ; BR IF END OF LIST
 001322  122762  000000G 000000G            CMPB    #CS.DST,C.STS(R2) ; IS THIS A DESTINATION DESCRIPTOR ?
 001330  001371                             BNE     30$             ; BR IF NO
 001332  020462  000000G                    CMP     R4,D$STYP(R2)   ; CHECK DESTINATION TYPE
 001336  001366                             BNE     30$             ; IF NE, WRONG TYPE
 001340  032767  000000G 000000G            BIT     #XS.KND,$OPRND  ; WILDCARD DESTINATIONS ?
 001346  001010                             BNE     40$             ; BR IF YES
 001350  026762  000000G 000000G            CMP     $DST,D$SNAM(R2) ; COMPARE FIRST WORD OF NAME
 001356  001356                             BNE     30$             ; BR IF WRONG DESTINATION
 001360  026762  000002G 000002G            CMP     $DST+2,D$SNAM+2(R2) ; COMPARE SECOND PART OF NAME
 001366  001352                             BNE     30$             ; IF NE, KEEP LOOKING
 001370  005203                     40$:    INC     R3              ; INDICATE FOUND OR WILDCARD

                                    ;
                                    ; CHECK FOR SPECIAL DESTINATIONS
                                    ;
 001372  012700  177774'                    MOV     #SPCDST-4,R0    ; POINT TO SPECIAL DESTINATION TABLE
 001376  005720                     50$:    TST     (R0)+           ; POINT TO NEXT SPECIAL DESTINATION
 001400  005720                     60$:    TST     (R0)+           ;  ...
 001402  001415                             BEQ     70$             ; IF EQ, NO MORE TO CHECK
 001404  020420                             CMP     R4,(R0)+        ; ELSE COMPARE DESTINATION TYPES
 001406  001373                             BNE     50$             ; IF NE, WRONG DESTINATION TYPE
 001410  022062  000000G                    CMP     (R0)+,D$SNAM(R2); COMPARE FIRST PART
```

```
                .TITLE  PURREM - PURGE REMOTE NODE
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1983, 1984, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - PERFORM PURGE COMMAND
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;

```

```
                                                                   G  8
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE               REFERENCES

S$ADD      000006            #5-85        7-221
S$CIR      000022            #5-88
S$DEV      000032            #5-89
S$DPA      000050            #5-91
S$DPC      000054            #5-92
S$FLG      000004            #5-84        7-238       *8-286
S$HAD      000012            #5-87
S$HST      000010            #5-86
S$LEN    = 000056            #5-93        7-250
S$PSS      000034            #5-90
S.LENX   = ******   GX        7-253       7-254
WC.NOD   = ******   GX        6-168       6-186        6-191
$BYTMN   = ******   GX       *6-183      *7-253      *10-360
$BYTXT   = ******   GX       *6-184      *7-254      *10-359
$CFERR   = ******   GX        6-197
$FILHD   = ******   GX        10-362
$NODAD   = ******   GX        6-171
$OPRND   = ******   GX        6-177       7-229       *7-255      8-284      9-317
$PUREM     000006 RG         #6-157
$RLBF    = ******   GX        6-182       7-252        7-261      10-371
$TMLST   = ******   GX        6-158       7-211      *11-394
$WILD    = ******   GX        6-168       6-186        6-191
$XXAVL   = ******   GX        6-181       7-251        7-260      10-370
.CONER   = ******   GX        7-266       10-374
```

```
                                                                   H  8
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 3
```

```
Table of contents
```

```
     5-  111     $SECNT - SET CONTROLLER PARAMETERS
     6-  189     UPDDDM - UPDATE THE DDM PRIORITY
```

```
                                        .SBTTL  LOCAL DATA
                                ;
                                ;  ERROR TEXT
                                ;
                                        .NLIST  BEX
         000002                         EX$ERR  = 2
 000000     002      103      106  ERRNSD: .ASCIZ  <EX$ERR>/CFE -- DTE not in configuration/
 000041     002      103      106  ERRDTE: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: DTE/
 000100     002      103      106  ERRNUM: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: NUMBER/
                                        .LIST   BEX
                                        .EVEN
```

```
               000005                                 .REPT    5
                                                      MOV      (R0)+,(R1)+         ; MOVE IT
                                                      .ENDR
        000354 032767  000000G 000000G 40$:           BIT      #XP.COU,$SWICH      ; COUNTER TIMER SPECIFIED ?
        000362 001403                                 BEQ      50$                 ; IF EQ, NO
        000364 016765  000000G 000000G                MOV      $CT,D$TCT(R5)       ; ELSE UPDATE IT
        000372 032767  000000G 000000G 50$:           BIT      #XP.MCI,$SWICH      ; MAXIMUM CIRCUITS ?
        000400 001415                                 BEQ      60$                 ; IF EQ, NO
        000402 016700  000000G                        MOV      $MCI,R0             ; GET NEW ALLOCATION IN ENTRIES
        000406 166500  000000G                        SUB      D$THSH(R5),R0       ; CALCULATE DIFFERENCE
        000412 006300                                 ASL      R0                  ; CONVERT
        000414 006300                                 ASL      R0                  ;  TO BYTES
        000416 060067  000000G                        ADD      R0,$BYTMN           ; UPDATE MINIMUM BYTE ALLOCATION
        000422 060067  000000G                        ADD      R0,$BYTXT           ; UPDATE ACTUAL BYTE ALLOCATION
        000426 016765  000000G 000000G                MOV      $MCI,D$THSH(R5)     ; UPDATE IT
        000434 032767  000000G 000000G 60$:           BIT      #XP.STA,$SWICH      ; CHANGE OF STATE ?
        000442 001402                                 BEQ      70$                 ; IF EQ, NOPE
        000444 004767  000406                         CALL     DSTATE              ; ELSE CHANGE DTE STATE
        000450 032767  000000G 000000G 70$:           BIT      #XP.LIN,$SWICH      ; CHANGE OF LINE ?
        000456 001402                                 BEQ      80$                 ; IF EQ, NO
        000460 004767  000122                         CALL     SETLIN              ; UPDATE LINE
        000464 032767  000000G 000000G 80$:           BIT      #XP.CHN,$SWICH      ; CHANNEL LIST SPECIFIED?
        000472 001403                                 BEQ      90$                 ; IF EQ, NO
        000474 004767  000174                         CALL     SETCHN              ; CREATE CHANNEL LIST
        000500 103405                                 BCS      101$                ; IF CS, ALLOCATION FAILURE
        000502 032767  000000G 000000G 90$:           BIT      #XP.KND,$OPRND      ; KNOWN DTES ?
        000510 001213                                 BNE      10$                 ; IF NE, CONTINUE
        000512 000207                  99$:           RETURN
                                       ;
                                       ; ERRORS
                                       ;
        000514                         101$:          ERROR$   #ERRALL             ; ALLOCATION ERROR
        000524 000404                                 BR       110$
        000526                         102$:          ERROR$   #ERRPMI             ; PARAMETER MISSING
        000536 000261                  110$:          SEC
        000540 000207                                 RETURN
```

```
                                        .SBTTL  MACRO CALLS

                                ;****
                                ; LIBRARY MACRO CALLS
                                ;****


                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                        .MCALL  FLTDF$

   000000                               FLTDF$                          ; DEFINE FILTER BLOCK SYMBOLS
```

Table of contents

```
     4-    53    MACRO DEFINITIONS
     5-    82    LOCAL DATA
     6-   109    $SEXSV- DEFINE MODULE X25-SERVER
     6-   110    $SEX29- DEFINE MODULE X29-SERVER
     7-   190    SETDST - SET DST$DF PARAMETERS
     8-   217    SETDSA - SET DSA$DF PARAMETERS
     9-   263    SETDSC - SET DSC$DF PARAMETERS
    10-   333    XSVCHA - UPDATE X25-SERVER CHARACTERISTICS
    11-   371    X29CHA - UPDATE X29-SERVER CHARACTERISTICS
    12-   400    CHKDST  - CHECK FOR DESTINATION TEMPLATE
    13-   434    ALCDST - ALLOCATE A DESTINATION DESCRIPTOR
    14-   498    $SEXAC - DEFINE MODULE X25-ACCESS
    15-   572    TSTPSI - TEST FOR PSI IN SYSTEM
    16-   590    RQCB - REQUEST CORE BLOCK
```

```
                                              .SBTTL  ALCDST - ALLOCATE A DESTINATION DESCRIPTOR
                                      ;+
                                      ; ** ALCDST - ALLOCATE A DESTINATION DESCRIPTOR
                                      ;
                                      ; INPUT           R3 = ADDRESS TO LINK NEW BLOCK
                                      ;                 R4 = RAD50 DESTINATION TYPE (X25 OR X29)
                                      ;                 $OPRND = PARSED PARAMETERS
                                      ;                 $XXAVL = FREE LIST
                                      ;
                                      ; OUTPUT          R2 = ADDRESS OF NEW TEMPLATES.
                                      ;                 THE NEW TEMPLATES ARE INITIALIZED AND
                                      ;                 LINKED INTO THE LIST.
                                      ;
                                      ; REGISTERS       R0, R1 ARE CLOBBERED.
                                      ;-

002226                                ALCDST:
002226  032767  000000G 000000G               BIT     #XS.DST,$OPRND  ; DESTINATION NAME PRESENT ?
002234  001503                                BEQ     40$             ; IF EQ, ERROR
002236  032767  000000C 000000G               BIT     #XS.OBJ!XS.OBI,$OPRND ; OBJECT PRESENT ?
002244  001504                                BEQ     50$             ; IF EQ, ERROR
002246  032767  000000G 000000G               BIT     #XS.PRI,$OPRND  ; PRIORITY SPECIFIED ?
002254  001006                                BNE     10$             ; IF NE, SPECIFIED
002256  052767  000000G 000000G               BIS     #XS.PRI,$OPRND  ; SET PRIORITY OPTION
002264  012767  000177  000000G               MOV     #DEFPRI,$PRI    ; SET DEFAULT PRIORITY

002272  012700  000000G               10$:    MOV     #$XXAVL,R0      ; POINT TO FREE LIST
002276  012701  000000C                       MOV     #<D$SLEN+3&^C3>,R1 ; DST$DF SIZE ROUNDED UP
002302  062701  000000C                       ADD     #<D$ALEN+3&^C3>,R1 ; DSA$DF SIZE ROUNDED UP
002306  062701  000000C                       ADD     #<D$CLEN+3&^C3>,R1 ; DSC$DF SIZE ROUNDED UP
002312  004767  000474                        CALL    RQCB            ; REQUEST CORE BLOCK
002316  103445                                BCS     30$             ; IF CS, ALLOCATION FAILURE
002320  010002                                MOV     R0,R2           ; SAVE POINTER
002322  062767  000000G 000000G               ADD     #D$FLNX,$BYTMN  ; UPDATE MINIMUM ALLOCATION
002330  062767  000000G 000000G               ADD     #D$FLNX,$BYTXT  ; UPDATE ACTUAL ALLOCATION
002336  010010                                MOV     R0,(R0)         ; SET TO SELF
002340  062720  000000C                       ADD     #D$SLEN+3&^C3,(R0)+     ; SET LINK TO DSA$DF
002344  012710  000000G                       MOV     #CS.DST,(R0)    ; SET UP TEMPLATE TYPE
002350  014000                                MOV     -(R0),R0        ; POINT TO DSA$DF
002352  010010                                MOV     R0,(R0)         ; SET LINK TO SELF
002354  062720  000000C                       ADD     #D$ALEN+3&^C3,(R0)+     ; SET LINK TO DSC$DF
002360  012710  000000G                       MOV     #CS.DSA,(R0)    ; MARK TEMPLATE TYPE
002364  014000                                MOV     -(R0),R0        ; POINT TO DSC$DF
002366  011310                                MOV     (R3),(R0)       ; LINK TO SUCCESSOR
002370  001002                                BNE     20$             ; IF NE, NOT LAST IN LIST
002372  010067  000002G                       MOV     R0,$TMLST+2     ; ELSE FIX LIST POINTER
002376  012760  000000G 000002        20$:    MOV     #CS.DSC,2(R0)   ; SET UP TEMPLATE TYPE
002404  010213                                MOV     R2,(R3)         ; LINK IN WITH PREDECESSOR
002406  016762  000000G 000000G               MOV     $DST,D$SNAM(R2) ; SET UP DESTINATION NAME
002414  016762  000002G 000002G               MOV     $DST+2,D$SNAM+2(R2) ; ...
002422  010462  000000G                       MOV     R4,D$STYP(R2)   ; SET UP DESTINATION TYPE
002426  000241                                CLC
002430  000207                                RETURN
                                      ;
                                      ; ERRORS
                                      ;
002432                                30$:    ERROR$  #ERRALL         ; ALLOCATION FAILURE
```

```
                                        .TITLE  SETNOD - SET EXECUTOR NODE OPTIONS
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; CFE - SET EXECUTOR NODE PARAMETERS
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  14-DEC-79
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;
                                ;
```

```
                .TITLE  SETNSP - SET NSP PARAMETERS
                .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SET NSP PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                 .SBTTL  .MOOBJ  - PARSE OBJECT STRING
                                         ;+
                                         ; **  - PARSE OBJECT STRING
                                         ;
                                         ; INPUTS        .TPARS VARAIBLES
                                         ;
                                         ; OUTPUT        $OBJ = TASKNAME OF OBJECT.
                                         ;-
 001176                                  .MOOBJ::
 001176  026727  000000G 000006                  CMP     .PSTCN,#6               ; CHECK SIZE OF PARSED STRING
 001204  101357                                  BHI     REJEC2                  ; IF HI, REJECT TRANSITION
 001206  016700  000000G                         MOV     .PSTPT,R0               ; POINT TO ASCII
 001212  010601                                  MOV     SP,R1                   ; ALLOW PERIODS
 001214  004767  000000G                         CALL    $CAT5                   ; CONVERT
 001220  010167  000000G                         MOV     R1,$OBJ                 ; LOAD UP FIRST PART
 001224  005067  000002G                         CLR     $OBJ+2                  ; ASSUME NO SECOND PART
 001230  026727  000000G 000003                  CMP     .PSTCN,#3               ; ANY MORE TO CONVERT ?
 001236  003404                                  BLE     10$                     ; IF LE, NOPE
 001240  004767  000000G                         CALL    $CAT5                   ; CONVERT REMAINDER
 001244  010167  000002G                         MOV     R1,$OBJ+2               ; SAVE IT
 001250  000207                          10$:    RETURN
```

H 2

```
PRSOBJ     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE              REFERENCES

ERRILN  = ******  GX        5-86        7-148
$CAT5   = ******  GX        6-111       6-115
$COPES  = ******  GX       *7-138      *7-145
$ERROR  = ******  GX       *5-86       *7-148
$TNAM   = ******  GX       *6-104      *6-112      *6-116
$TNUM   = ******  GX       *5-80
.OBCOP    000136 RG        #7-137
.OBNAM    000036 RG        #6-103
.OBNUM    000000 RG        #5-79
.PNUMB  = ******  GX        5-80        5-83        7-138       7-142
.PNUMH  = ******  GX        5-81        7-140
.PSTCN  = ******  GX        6-107       6-117
.PSTPT  = ******  GX        6-105
```

```
                                                .SBTTL  DEFINITIONS AND DATA AREA
                                        ;
                                        ; Define file descriptor offsets
                                        ;

                000036                          MXFSP   = 30.           ; Undivided Filespec length

        000000                                  .ASECT
                000000                          . = 0
        000000                                  .BLKB   1               ; Filespec string length
        000001                          F.DEV:  .BLKB   MXFSP           ; Filespec string (ASCII)
                000037                  F.LEN   = .
        000000                                  .PSECT

                                        ;
                                        ; DEFINE REMOTE TEMPLATE
                                        ;
        000037                                  .ASECT
                000000                          .=0
        000000                                  .BLKW   1               ; C.LNK
        000002                                  .BLKW   1               ; C.STS
        000004                          C.NAM:  .BLKW   2               ; REMOTE NODE NAME
        000010                          C.ADD:  .BLKW   1               ; REMOTE NODE ADDRESS
                000012                  C.LEN=.
        000000                                  .PSECT

                                        ;
                                        ; LOCAL DATA
                                        ;
        000000                          SCR:    .BLKB   16.             ; SCRATCH BUFFER

                                                .SBTTL  .RNIN2 - INITIALIZE REMOTE NODE OPTIONS
                                        ;+
                                        ; **-.RNIN2 - INITIALIZE REMOTE NODE OPTIONS
                                        ;
                                        ; INPUTS: NONE.
                                        ;
                                        ; OUTPUT: PARSED DATA AREA IS CLEARED.
                                        ;
                                        ;-
        000020                          .RNIN2::
        000020  012700  000000G                 MOV     #$RNHOS,R0      ; POINT TO START OF DATA AREA
        000024  005020                  10$:    CLR     (R0)+           ; CLEAR A WORD
        000026  020027  000000G                 CMP     R0,#$RNXXX      ; DONE ?
        000032  103774                          BLO     10$             ; NO
        000034  000207                          RETURN
```

```
**FILE**ID**PRSSYS




PPPPPPPP     RRRRRRRR       SSSSSSSS     SSSSSSSS   YY      YY     SSSSSSSS
PPPPPPPP     RRRRRRRR       SSSSSSSS     SSSSSSSS   YY      YY     SSSSSSSS
PP      PP   RR      RR   SS           SS           YY      YY   SS
PP      PP   RR      RR   SS           SS           YY      YY   SS
PP      PP   RR      RR   SS           SS             YY  YY     SS
PP      PP   RR      RR   SS           SS             YY  YY     SS
PPPPPPPP     RRRRRRRR       SSSSSS       SSSSSS         YY         SSSSSS
PPPPPPPP     RRRRRRRR       SSSSSS       SSSSSS         YY         SSSSSS
PP           RR  RR               SS           SS       YY               SS
PP           RR  RR               SS           SS       YY               SS
PP           RR    RR             SS           SS       YY               SS       ....
PP           RR    RR             SS           SS       YY               SS       ....
PP           RR      RR   SSSSSSSS     SSSSSSSS         YY       SSSSSSSS         ....
PP           RR      RR   SSSSSSSS     SSSSSSSS         YY       SSSSSSSS         ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
PRSSYS     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE            REFERENCES

CS.ROU   = ******  GX      10-221
C.STS    = ******  GX      10-221
ERRILN   = ******  GX      7-143      8-169     9-196     10-229    11-255    12-283
$BYT     = ******  GX     *7-139
$CAT5    = ******  GX      5-84       5-87      6-113     6-116
$CCBNM   = ******  GX     *8-161
$ERROR   = ******  GX     *7-143     *8-169    *9-196    *10-229   *11-255   *12-283
$PARNA   = ******  GX     *5-81      *5-85     *5-88
$POLNA   = ******  GX     *6-110     *6-114    *6-117
$RDBNM   = ******  GX     *9-187      12-278
$RDBSZ   = ******  GX     *10-214
$SDBNM   = ******  GX     *11-247
$THRSH   = ******  GX     *12-273
$TMLST   = ******  GX      10-218
.PNUMB   = ******  GX      7-138      8-161     8-164     8-166     9-187     9-190     10-214    10-224    10-226
                           11-247     11-250    11-252    12-273    12-276    12-280
.PNUMH   = ******  GX      8-162      9-188     10-215    11-248    12-274
.PSTCN   = ******  GX      5-89       6-118
.PSTPT   = ******  GX      5-82       6-111
.SYBYT     000124  RG     #7-137
.SYCBN     000156  RG     #8-160
.SYPAR     000000  RG     #5-80
.SYPOL     000052  RG     #6-109
.SYRBN     000226  RG     #9-186
.SYRBS     000266  RG     #10-213
.SYSBN     000364  RG     #11-246
.SYTHR     000434  RG     #12-272
```

```
001414  001371                             BNE     60$               ; IF NE, NO MATCH - CONTINUE
001416  021062  000002G                    CMP     (R0),D$SNAM+2(R2) ; COMPARE SECOND PART
001422  001366                             BNE     60$               ; IF NE, NO MATCH - CONTINUE
001424  032767  000000G 000000G            BIT     #XS.KND,$OPRND    ; KNOWN DESTINATIONS ?
001432  001035                             BNE     100$              ; IF NE, YES - JUST IGNORE THIS ONE
001434  000451                             BR      130$              ; ELSE ERROR - NO SUCH DESTINATION

001436  012700  000000G            70$:    MOV     #D$FLNX,R0        ; LENGTH OF FIXED PART OF DESTINATION BLOCK
001442  012701  000000G                    MOV     #D$SLEN,R1        ; LENGTH TO RETURN
001446  004767  000412                     CALL    DEALC             ; DEALLOCATE DESTINATION BLOCK
001452  122762  000000G 000000G            CMPB    #CS.DSA,C.STS(R2) ; EXPECTING DESTINATION ADDRESS NEXT
001460  001031                             BNE     120$              ; IF NE, ERROR
001462  016200  000000G                    MOV     D$ADLN(R2),R0     ; LENGTH OF DTE ADDRESS
001466  012701  000000G                    MOV     #D$ALEN,R1        ; LENGTH TO RETURN
001472  004767  000366                     CALL    DEALC             ; DEALLOCATE THE ADDRESS BLOCK
001476  122762  000000G 000000G    80$:    CMPB    #CS.DSC,C.STS(R2);  EXPECTING DESTINATION CALL INFORMATION
001504  001017                             BNE     120$              ; IF NE, BIG TROUBLE
001506  116200  000000G                    MOVB    D$CVLN(R2),R0     ; LENGTH OF CALL MASK AND CALL VALUE
001512  012701  000000G                    MOV     #D$CLEN,R1        ; LENGTH TO RETURN
001516  004767  000342                     CALL    DEALC             ; DEALLOCATE THE BLOCK
001522  010502                     90$:    MOV     R5,R2             ; RESUME WHERE WE LEFT OFF
001524  000673                             BR      30$               ; CONTINUE

001526  032767  000000G 000000G    100$:   BIT     #XS.KND,$OPRND    ; KNOWN DESTINATIONS ?
001534  001002                             BNE     110$              ; BR IF YES
001536  005703                             TST     R3                ; WAS DESTINATION FOUND ?
001540  001407                             BEQ     130$              ; IF EQ, NO - ERROR
001542  000207                     110$:   RETURN

                                   ;
                                   ; ERRORS
                                   ;
001544                             120$:   ERROR$  #ERRDST           ; INVALID DESTINATION SEQUENCE
001554  000167  000000G                    JMP     .CONER            ; THIS IS SERIOUS !
001560                             130$:   ERROR$  #ERRNSD           ; NO SUCH DESTINATION
001570  000207                             RETURN

                                           .DSABL  LSB
```

```
                .SBTTL  MACRO DEFINITIONS

        ;****
        ; LOCAL MACROS
        ;****

                .MACRO  PRINT   TEXT
                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                CALL    $PRINT          ; PRINT MESSAGE
                .ENDM   PRINT

                .MACRO  ERROR$  TEXT
        ;
        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
        ; MESSAGE STRING.
        ;
                .IF DIF <TEXT><R0>
                MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                .ENDC
                CALL    $CFERR                  ; PRINT ERROR MESSAGE
                .ENDM   ERROR$

                .MCALL  SAVRG,RESRG
```

```
PURREM     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20     PAGE 3

MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

ERROR$           #4-56       6-197
PRINT            #4-51
RESRG            #4-67       7-263       9-326
SAVRG            #4-67       7-210       9-313
```

FILEID**SETBUF

```
                        .TITLE  SETCNT - SET CONTROLLER OPTIONS
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - SET CONTROLLER PARAMETERS
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                                .SBTTL  $SECUG - SET CLOSED USER GROUP PARAMETERS
                                        ;+
                                        ; *** - $SECUG - SET CLOSED USER GROUP PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       THE CUG$DF TEMPLATE IS UPDATED
                                        ;
                                        ;-

000142                                  $SECUG::
                                        ;
                                        ; LOOK FOR GROUP DESCRIPTOR
                                        ;
000142  012705  000000G                         MOV     #$IMLST,R5        ; POINT TO TEMPLATES
000146  010504                          10$:    MOV     R5,R4             ; SAVE PREVIOUS ADDRESS
000150  011505                                  MOV     (R5),R5           ; NEXT ONE
000152  001421                                  BEQ     15$               ; IF EQ, END OF LIST
000154  122765  000000G 000000G                 CMPB    #CS.CUG,C.STS(R5) ; CHECK TYPE
000162  001371                                  BNE     10$               ; BR IF WRONG TYPE
000164  032767  000000G 000000G                 BIT     #XP.KND,$OPRND    ; KNOWN GROUPS ?
000172  001057                                  BNE     20$               ; IF NE, YES
000174  026765  000000G 000000G                 CMP     $CUG,C$UNAM(R5)   ; COMPARE NAMES
000202  001361                                  BNE     10$               ; BR IF WRONG NAME
000204  026765  000002G 000002G                 CMP     $CUG+2,C$UNAM+2(R5);  ...
000212  001355                                  BNE     10$               ;  ...
000214  000446                                  BR      20$               ; BR IF THEY MATCH
                                        ;
                                        ; CREATE A NEW GROUP
                                        ;
000216  032767  000000G 000000G         15$:    BIT     #XP.KNG,$OPRND    ; KNOWN GROUPS ?
000224  001144                                  BNE     60$               ; IF NE, YES - DONE
000226  032767  000000G 000000G                 BIT     #XP.DTA,$SWICH    ; DTE ADDRESS PARSED ?
000234  001557                                  BEQ     70$               ; BR IF NO
000236  032767  000000G 000000G                 BIT     #XP.NUM,$SWICH    ; GROUP NUMBER PARSED?
000244  001560                                  BEQ     72$               ; BR IF NO
000246  012700  000000G                         MOV     #$XXAVL,R0        ; POINT TO FREE LIST
000252  012701  000000G                         MOV     #C$ULEN,R1        ; LENGTH OF A GROUP TEMPLATE
000256  004767  000000G                         CALL    $RQBF             ; REQUEST CORE BLOCK
000262  103556                                  BCS     75$               ; IF CS, ALLOCATION FAILURE
000264  062767  000000G 000000G                 ADD     #G$LENX,$BYTMN    ; UPDATE BYTE MINIMUM
000272  062767  000000G 000000G                 ADD     #G$LENX,$BYTXT    ; UPDATE BYTE ALLOCATION
000300  012760  000000G 000000G                 MOV     #CS.CUG,C.STS(R0); MARK IT AS A GROUP DESCRIPTOR
000306  016760  000000G 000000G                 MOV     $CUG,C$UNAM(R0)   ; STUFF NAME INTO TEMPLATE
000314  016760  000002G 000002G                 MOV     $CUG+2,C$UNAM+2(R0);  ...
000322  005060  000000G                         CLR     C$UFLG(R0)        ; CLEAR FLAGS
000326  010005                                  MOV     R0,R5             ; COPY POINTER
000330  005004                                  CLR     R4                ; REMEMBER NOT TO UNLINK
000332  032767  000000G 000000G         20$:    BIT     #XP.NUM,$SWICH    ; CHANGE GROUP NUMBER?
000340  001403                                  BEQ     25$               ; IF EQ, NO
000342  016765  000000G 000000G                 MOV     $NUM,C$UCUG(R5)   ; UPDATE TEMPLATE
000350  032767  000000G 000000G         25$:    BIT     #XP.BIL,$SWICH    ; TYPE BILATERAL?
000356  001406                                  BEQ     30$               ; BR IF NO
000360  152765  000000G 000000G                 BISB    #CU$BIL,C$UFLG(R5) ; MARK AS BILATERAL GROUP
```

```
                                         .SBTTL  INSDTE - INSERT DTE TEMPLATE IN LIST
                                 ;+
                                 ; ** INSDTE - INSERT DTE TEMPLATE IN LIST
                                 ;
                                 ; INPUTS        R5 = ADDRESS OF TEMPLATE
                                 ;               $PSIPT = POINTER TO PSI PART OF TEMPLATE LIST
                                 ;
                                 ; OUTPUT        THE TEMPLATE IS LINKED INTO THE LIST.
                                 ;
                                 ; REGISTERS     R0,R1 AND R2 ARE CORRUPTED.
                                 ;-

000542                           INSDTE:
000542  012700  000000G                  MOV     #$PSIPT,R0         ; POINT TO PSI TEMPLATES
                                 ;
                                 ; SKIP ALL TEMPLATES WHICH CAN PRECEDE A DTE TEMPLATE
                                 ;
000546  010001                   10$:    MOV     R0,R1              ; SAVE PREVIOUS POINTER
000550  011000                           MOV     (R0),R0            ; NEXT TEMPLATE
000552  001407                           BEQ     30$                ; IF EQ, LINK HERE
000554  012702  000000'                  MOV     #PRDLST,R2         ; POINT TO START OF PRECEDE LIST
000560  126022  000000G          20$:    CMPB    C.STS(R0),(R2)+    ; CHECK TEMPLATE TYPE
000564  001770                           BEQ     10$                ; IF EQ, LEGAL PREDECESSOR
000566  105712                           TSTB    (R2)               ; CHECK FOR END OF PRECEDE LIST
000570  001373                           BNE     20$                ; IF NE, KEEP LOOKING
                                 ;
                                 ; LINK TEMPLATE INTO LIST
                                 ;
000572  011115                   30$:    MOV     (R1),(R5)          ; POINT NEW TEMPLATE TO SUCCESSOR
000574  001002                           BNE     40$                ; IF NE, NOT LAST
000576  010567  000002G                  MOV     R5,$TMLST+2        ; UPDATE POINTER TO LAST TEMPLATE
000602  010511                   40$:    MOV     R5,(R1)            ; POINT PREDECESSOR TO NEW TEMPLATE
000604  000207                           RETURN                     ; DONE

```

```
                                                         .SBTTL  LOCAL DATA
                                                 ;****
                                                 ; LOCAL DATA
                                                 ;****

                                                 ;
                                                 ; SINK TYPE TABLE
                                                 ;
 000000  000001                                  SNKTBL: .WORD   FF.CON          ; CONSOLE SINK TYPE
 000002  000002                                          .WORD   FF.FIL          ; FILE SINK TYPE
 000004  000004                                          .WORD   FF.MON          ; MONITOR SINK TYPE
 000006  000000                                          .WORD   0               ; END OF TABLE

                                                         .ENABL  LC
         000002                                          EX$ERR = 2
         000000                                          EX$WAR = 0
 000010     002        103        106        ERRBUF: .ASCIZ  <EX$ERR>/CFE -- Buffer allocation failure/
 000052     002        103        106        ERRLOG: .ASCIZ  <EX$ERR>/CFE -- No logging state definition/
 000116     000        103        106        ERREVT: .ASCIZ  <EX$WAR>/CFE -- Events cannot be filtered/
                                                         .EVEN
```

```
        .TITLE  SETMOD - DEFINE MODULE
        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - DEFINE MODULE
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  08-NOV-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
    002442  000411                          BR      60$
    002444                          40$:    ERROR$  #ERRDST         ; MISSING DESTINATION ID
    002454  000404                          BR      60$
    002456                          50$:    ERROR$  #ERROBJ         ; MISSING OBJECT PARAMETER
    002466  000261                  60$:    SEC
    002470  000207                          RETURN
```

```

                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                ;****
                                ; LOCAL DATA
                                ;****

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .NLIST  BEX
                                        .ENABL  LC
        000002                          EX$ERR = 2
000000     002     103     106  ERRNOD: .ASCIZ  <EX$ERR>/CFE -- No node definition/
                                        .EVEN

                                ;
                                ; DEFINE NODE TEMPLATE
                                ;
000000                                  .ASECT
        000000                          .=0
000000                                  .BLKW   1               ; C.LNK
000002                                  .BLKW   1               ; C.STS
000004                          C.NAM:  .BLKW   2               ; NODE NAME (RAD50)
000010                          C.NID:  .BLKW   1               ; NODE ID
000012                                  .BLKB   32.             ; LENGTH AND STRING
000052                          C.NUM:  .BLKW   1               ; NODE ADDRESS
000054                          C.HOS:  .BLKW   1               ; HOST NODE ADDRESS
        000056                  C.LEN=.
```

```
                        ;****
                        ; LIBRARY MACRO CALLS
                        ;****

                                .MCALL  NSFDF$,LLCTP$,DECTP$

000000                          DECTP$  LIST            ; DEFINE DEC$DF TEMPLATE OFFSETS
                        ;
                        ; DEFINE DEC TEMPLATE
                        ;
000000                          .ASECT
          000000                .=0
000000                          .BLKW   1               ; C.LNK
000002                          .BLKW   1               ; C.STS
000004                  C.SEG:  .BLKW   1               ; SEGMENT SIZE
000006                  C.NDC:  .BLKW   1               ; MAXIMUM NODE COUNTERS
000010                  C.LKS:  .BLKW   1               ; MAXIMUM LOGICAL LINKS
000012                  C.INCT: .BLKB   1               ; INCOMING TIMER
000013                  C.OUTT: .BLKB   1               ; OUTGOING TIMER
000014                  C.INAC: .BLKB   1               ; INACTIVITY TIMER
000015                  C.RETF: .BLKB   1               ; RETRANSMIT FACTOR
000016                  C.DELF: .BLKB   1               ; DELAY FACTOR
000017                  C.DELW: .BLKB   1               ; DELAY WEIGHT
000020                  C.IPL:  .BLKB   1               ; INPUT PACKET LIMITER
000021                  C.LST:  .BLKB   1               ; LINK SERVICE THRESHOLD
          000022        CD.LEN=.
000000                          .PSECT
000000                          LLCTP$  LIST            ; DEFINE LLC$DF TEMPLATE OFFSETS
                        ;
                        ; DEFINE LLC TEMPLATE
                        ;
000022                          .ASECT
          000000                .=0
000000                          .BLKW   1               ; C.LNK
000002                          .BLKW   1               ; C.STS
000004                  C.NAM:  .BLKW   1               ; LLC PROCESS NAME
000006                  C.FLG:  .BLKW   1               ; LLC FLAGS WORD
000010                  C.PRI:  .BLKB   1               ; LLC PROCESS PRIORITY
000011                  C.LIN:  .BLKB   1               ; NUMBER OF LINES ASSIGNED TO THIS LLC
000012                  C.DEV:  .BLKW   1               ; PSEUDO DEVICE NAME FOR THIS LLC
000014                  C.NLT:  .BLKB   1               ; NUMBER OF LINE TABLE TO ALLOCATE
000015                  C.EXT:  .BLKB   1               ; PROCESS EXTENSION IN BLOCKS
000016                  C.CTM:  .BLKW   1               ; COUNTER TIMER
          000020        CL.LEN=.
000000                          .PSECT
          000012        C.NLTD = 12                     ; NLT FOR DDM/DLC PROCESSES
          000013        C.NCTD = 13                     ; NCT FOR DDM/DLC PROCESSES
000000                          NSFDF$                  ; DEFINE NSP FEATURES MASK BITS

                        ;****
                        ; LOCAL MACROS
                        ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT
```

```
    562                                                 .SBTTL  .MONNA  - PARSE NETWORK NAME
    563                                         ;+
    564                                         ; ** .MONNA - PARSE NETWORK NAME
    565                                         ;
    566                                         ; INPUTS        .TPARS VARIABLES.
    567                                         ;
    568                                         ; OUTPUT        $NNAM = PARSED NETWORK NAME
    569                                         ;               THE TRANSITION IS ACCEPTED/REJECTED.
    570                                         ;
    571                                         ;-
    572 001252                                  .MONNA::
    573 001252  026727  000000G 000006                  CMP     .PSTCN,#6               ; CHECK SIZE OF PARSED STRING
    574 001260  101331                                  BHI     REJEC2                  ; IF HI, REJECT TRANSITION
    575 001262  016700  000000G                         MOV     .PSTPT,R0               ; POINT TO ASCII
    576 001266  005001                                  CLR     R1                      ; ALLOW PERIODS
    577 001270  004767  000000G                         CALL    $CAT5                   ; CONVERT
    578 001274  010167  000000G                         MOV     R1,$NNAM                ; LOAD UP FIRST PART
    579 001300  005067  000002G                         CLR     $NNAM+2                 ; ASSUME NO SECOND PART
    580 001304  026727  000000G 000003                  CMP     .PSTCN,#3               ; ANY MORE TO CONVERT ?
    581 001312  003404                                  BLE     10$                     ; IF LE, NOPE
    582 001314  004767  000000G                         CALL    $CAT5                   ; CONVERT REMAINDER
    583 001320  010167  000002G                         MOV     R1,$NNAM+2              ; SAVE IT
    584 001324  000207                          10$:    RETURN
    585
    586         000001                                  .END
```

```
PRSOBJ     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME      REFERENCES
PRINT           #4-59
```

```
                                                .SBTTL  .RNCHK - CHECK REMOTE NODE OPTIONS
                                        ;+
                                        ; **-.RNCHK-CHECK REMOTE NODE OPTIONS
                                        ;
                                        ; INPUTS:       $OPRND  = PARSED OPTIONS
                                        ;               $NODAD  = NODE ADDRESS
                                        ;               $NODNA  = NODE NAME
                                        ;
                                        ; OUTPUT:       TRANSITION REJECTED IF:
                                        ;               - NODE IS EXECUTOR AND REMOTE NODE OPTIONS SPECIFIED
                                        ;               - NODE IS REMOTE AND THERE IS NO NAME FOR IT AND
                                        ;                 THERE IS NO NAME SPECIFIED IN THE CURRENT COMMAND.
                                        ;
                                        ;-

000036                                  .RNCHK::
                                        ;
                                        ; NODE NAME PARSED: CHECK AGAINST EXECUTOR'S NAME
                                        ;
000036  032767  000000G 000000G                 BIT     #RN.NAM,$OPRND  ; PARSED A NODE NAME ?
000044  001415                                  BEQ     10$             ; IF EQ, PARSED AN ADDRESS
000046  026767  000000G 000000G                 CMP     $NODNA,$NTNAM   ; EXECUTOR NAME ?
000054  001037                                  BNE     30$             ; IF NE, NO
000056  026767  000002G 000002G                 CMP     $NODNA+2,$NTNAM+2 ; EXEC NAME  ?
000064  001033                                  BNE     30$             ; IF NE, NO
000066  000421                                  BR      16$             ; ERROR
                                        ;
                                        ; NODE ADDRESS PARSED: MAKE SURE ADDRESS IS NOT THE EXECUTOR'S
                                        ; AND BELONGS TO A REMOTE NODE ALREADY KNOWN TO THE SYSTEM.
                                        ;
000070  026767  000000G 000000G                 CMP     $NODAD,$NTADD   ; EXECUTOR NODE ?
000076  001415                                  BEQ     16$             ; YES - BAD PARAMS

000100  012700  000000G                 10$:    MOV     #$TMLST,R0      ; GET START OF TEMPLATES

000104  011000                          15$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
000106  001415                                  BEQ     17$             ; IF EQ, END OF TEMPLATES
000110  122760  000000G 000000G                 CMPB    #CS.REM,C.STS(R0) ; ELSE, IS THIS A REMOTE TEMPLATE ?
000116  001372                                  BNE     15$             ; IF NE, NO .. KEEP LOOKING
000120  026760  000000G 000010                  CMP     $NODAD,C.ADD(R0) ; DO ADDRESSES MATCH ?
000126  001366                                  BNE     15$             ; IF NE, NO
000130  000411                                  BR      30$             ; NODE OK

000132  012767  000000G 000000G 16$:    MOV     #ERRPNA,$ERROR  ; INVALID PARAMETERS FOR EXEC
000140  000403                                  BR      20$

000142  012767  000000G 000000G 17$:    MOV     #ERRNOD,$ERROR  ; NODE NOT IN CONFIGURATION

000150  062716  000002                  20$:    ADD     #2,(SP)

000154  000207                          30$:    RETURN

```

Table of contents

```
     5-    65     .SYPAR - DYNAMIC PARTITION NAME ACTION ROUTINE
     6-    94     .SYPOL - POOL NAME ACTION ROUTINE
     7-   124     .SYBYT - POOL SINGLE WORD ALLOCATION EXTENSION
     8-   146     .SYCBN - SET NUMBER OF CCB'S
     9-   172     .SYRBN - SET NUMBER OF RDB'S
    10-   199     .SYRBS - SET SIZE OF EACH RDB
    11-   232     .SYSBN - SET NUMBER OF SDB'S
    12-   258     .SYTHR - SET MINIMUM NUMBER OF RDB'S FOR POOL
```

I 5

```
PRSSYS     CREATED BY  MACRO  ON 29-JUN-85 AT 05:20      PAGE 2
MACRO CROSS REFERENCE                                    CREF    04.00
MACRO NAME       REFERENCES
PRINT            #4-60
```

J 5

```
                                                .SBTTL  XSVCTM - PURGE X25-SERVER COUNTER TIMER
                                        ;+
                                        ; ** XSVCTM - PURGE X25-SERVER COUNTER TIMER
                                        ;
                                        ; INPUTS        NONE.
                                        ;
                                        ; OUTPUT        THE X25-SERVER COUNTER TIMER IS PURGED.
                                        ;-

001572                                  XSVCTM:
001572  012700  000000G                         MOV     #$TMLST,R0      ; POINT TO LISTHEAD
001576  011000                          10$:    MOV     (R0),R0         ; NEXT TEMPLATE
001600  001416                                  BEQ     101$            ; BR IF ERROR
001602  122760  000000G 000000G                 CMPB    #CS.LLC,C.STS(R0) ; CHECK TYPE
001610  001372                                  BNE     10$             ; BR IF NOT LLC
001612  026027  000000G 055430                  CMP     L$LNAM(R0),#<^RNW > ; CHAECK LLC NAME
001620  001366                                  BNE     10$             ; BR IF WRONG LLC
001622  005060  000000G                         CLR     L$LCTM(R0)      ; PURGE COUNTER TIMER
001626  052760  000000G 000000G                 BIS     #CS.MOD,C.STS(R0) ; MARK TEMPLATED FOR UPDATE
001634  000207                                  RETURN
                                        ;
                                        ; ERRORS
                                        ;
001636                                  101$:   ERROR$  #ERRXSV         ; NO X25-SERVER DATABASE
001646  000207                                  RETURN
```

```
                        .SBTTL  LOCAL DATA
                        .NLIST  BEX

                ;****
                ; LOCAL DATA
                ;****

                ;
                ; DEFINE SER$DF TEMPLATE
                ;
000000                  .ASECT
        000000          . = 0
000000                  .BLKW   1               ; C.LNK
000002                  .BLKW   1               ; C.FLG
000004          S$FLG:  .BLKW   1               ; option flags
000006          S$ADD:  .BLKW   1               ; Target node address    (16 bit node address)
000010          S$HST:  .BLKW   1               ; Host node address      (16 bit node address)
000012          S$HAD:  .BLKW   4               ; Hardware address       (RAD50)
000022          S$CIR:  .BLKW   4               ; Service circuit        (extended RAD50)
000032          S$DEV:  .BLKW   1               ; Service device         (extended RAD50)
000034          S$PSS:  .BLKW   6               ; Service password       (RAD50)
000050          S$DPA:  .BLKW   2               ; Dump address           (octal binary)
000054          S$DPC:  .BLKW   1               ; Dump count             (in k bytes)
        000056  S$LEN=.
000000                  .PSECT
                ;
                ; Flags definitions:
                ;
        000001  SF.ADD = 1                      ; Target address specified
        000002  SF.HST = 2                      ; Host address specified
        000004  SF.HAD = 4                      ; Hardware address specified
        000010  SF.CIR = 10                     ; Service circuit specified
        000020  SF.DEV = 20                     ; Service device specified
        000040  SF.PSS = 40                     ; Service password specified
        000100  SF.DPA = 100                    ; Dump address specified
        000200  SF.DPC = 200                    ; Dump count specified
        000400  SF.PH3 = 400                    ; Service node type is Phase 3

                ;
                ; Define FIL$DF template
                ;
000056                  .ASECT
        000000          . = 0
000000                  .BLKW   1               ; Link
000002                  .BLKW   1               ; Flags
000004          F$SEC:  .BLKW   1               ; Secondary loader file name
000006          F$TER:  .BLKW   1               ; Tertiary loader file name
000010          F$LD:   .BLKW   1               ; Load file name
000012          F$DIA:  .BLKW   1               ; Diagnostics file name
000014          F$DUM:  .BLKW   1               ; Dump file name
        000016  F$LEN=.
000000                  .PSECT
                ;
                ; Define file descriptor offsets
                ;

        000036          MXFSP   = 30.           ; Max filespec length
```

```
**FILE**ID**SETBUF





  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   BBBBBBBB     UU      UU   FFFFFFFFFF
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   BBBBBBBB     UU      UU   FFFFFFFFFF
SS           EE               TT       BB      BB   UU      UU   FF
SS           EE               TT       BB      BB   UU      UU   FF
SS           EE               TT       BB      BB   UU      UU   FF
SS           EE               TT       BB      BB   UU      UU   FF
  SSSSSS     EEEEEEEE         TT       BBBBBBBB     UU      UU   FFFFFFFF
  SSSSSS     EEEEEEEE         TT       BBBBBBBB     UU      UU   FFFFFFFF
        SS   EE               TT       BB      BB   UU      UU   FF
        SS   EE               TT       BB      BB   UU      UU   FF
        SS   EE               TT       BB      BB   UU      UU   FF               ....
        SS   EE               TT       BB      BB   UU      UU   FF               ....
SSSSSSSS     EEEEEEEEEE       TT       BBBBBBBB     UUUUUUUUUU   FF               ....
SSSSSSSS     EEEEEEEEEE       TT       BBBBBBBB     UUUUUUUUUU   FF               ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
     55                                              ;****
     56                                              ; LOCAL MACROS
     57                                              ;****
     58
     59                                                      .MACRO  PRINT   TEXT
     60                                                      MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
     61                                                      CALL    $PRINT          ; PRINT MESSAGE
     62                                                      .ENDM   PRINT
     63
     64                                                      .MACRO  ERROR$  TEXT
     65                                              ;
     66                                              ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
     67                                              ; MESSAGE STRING.
     68                                              ;
     69                                                      .IF DIF <TEXT><R0>
     70                                                      MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
     71                                                      .ENDC
     72                                                      CALL    $CFERR                  ; PRINT ERROR MESSAGE
     73                                                      .ENDM   ERROR$
     74
     75                                                      .MACRO  EPRINT  TEXT
     76                                              ;
     77                                              ; PRINT TEXT ON ERROR LUN
     78                                              ;
     79                                                      .IF DIF <TEXT><R0>
     80                                                      MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
     81                                                      .ENDC
     82                                                      CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
     83                                                      .ENDM   EPRINT
     84
     85                                              ;****
     86                                              ; LOCAL DATA
     87                                              ;****
     88
     89                                                      .ENABL  LC
     90                                                      .NLIST  BEX
     91                  000002                              EX$ERR = 2
     92 000000              002         103         106     ERRPRI: .ASCIZ  <EX$ERR>/CFE -- Illegal device priority/
     93                                                      .EVEN
     94                                                      .LIST   BEX
     95                                              ;
     96                                              ; DEFINE CONTROLLER TEMPLATE
     97                                              ;
     98 000000                                              .ASECT
     99                  000000                              .=0
    100 000000                                              .BLKW   1               ; C.LNK
    101 000002                                              .BLKW   1               ; C.STS
    102 000004                                      C.CNT:  .BLKW   1               ; CONTROLLER NUMBER
    103 000006                                      C.VEC:  .BLKW   1               ; VECTOR ADDRESS
    104 000010                                      C.CSR:  .BLKW   1               ; CSR ADDRESS
    105 000012                                      C.PRI:  .BLKW   1               ; CONTROLLER PRIORITY
    106 000014                                      C.URM:  .BLKW   1               ; UNIBUS RUN MASK
    107                  000016                      C.LEN=.
    108 000040                                              .PSECT
    109
```

```
000366  052765  000000G 000000G           BIS     #CS.FLG,C.STS(R5) ; REMEMBER FLAGS PRESENT
000374  032767  000000G 000000G  30$:     BIT     #XP.DTA,$SWICH    ; DTE ADDRESS ?
000402  001455                            BEQ     60$               ; BR IF NO
                                 ;
                                 ; LOOK FOR NEW DTE TEMPLATE
                                 ;
000404  012703  000000G                   MOV     #$TMLST,R3        ; POINT TO TEMPLATE LISTHEAD
000410  011303                   35$:     MOV     (R3),R3           ; NEXT
000412  001452                            BEQ     65$               ; BR IF NO SUCH DTE
000414  122763  000000G 000000G           CMPB    #CS.DTE,C.STS(R3);  CHECK TYPE
000422  001372                            BNE     35$               ; BR IF NOT DTE
000424  010302                            MOV     R3,R2             ; COPY TEMPLATE
000426  062702  000000G                   ADD     #D$TADR,R2        ; TEMPLATE DTE ADDRESS
000432  012701  000000G                   MOV     #$DTE,R1          ; PARSED DTE ADDRESS
000436  012700  000005                    MOV     #5,R0             ; LENGTH IN WORDS
000442  022122                   40$:     CMP     (R1)+,(R2)+       ; COMPARE
000444  001361                            BNE     35$               ; BR IF WRONG DTE
000446  005300                            DEC     R0                ; REDUCE WORDS REMAINING
000450  003374                            BGT     40$               ; BR IF MORE
                                 ;
                                 ; LINK GROUP IN AFTER NEW DTE
                                 ;
000452  005704                            TST     R4                ; ANYTHING TO UNLINK?
000454  001404                            BEQ     45$               ; BR IF NO
000456  011514                            MOV     (R5),(R4)         ; UNLINK GROUP
000460  001002                            BNE     45$               ; BR IF NOT END
000462  010567  000002G                   MOV     R5,$TMLST+2       ; ELSE MARK NEW END
000466  010304                   45$:     MOV     R3,R4             ; SAVE PREVIOUS ADDRESS
000470  011303                            MOV     (R3),R3           ; NEXT TEMPLATE
000472  001414                            BEQ     50$               ; BR IF END OF LIST
000474  122763  000000G 000000G           CMPB    #CS.CHN,C.STS(R3);  SKIP OVER CHANNEL BLOCKS
000502  001771                            BEQ     45$               ; ...
000504  122763  000000G 000000G           CMPB    #CS.PVC,C.STS(R3);  SKIP PVCS
000512  001765                            BEQ     45$               ; ...
000514  122763  000000G 000000G           CMPB    #CS.CUG,C.STS(R3);  SKIP OTHER GROUPS
000522  001761                            BEQ     45$               ; ...
000524  011415                   50$:     MOV     (R4),(R5)         ; LINK IN NEW SUCCESSOR
000526  001002                            BNE     55$               ; IF NE, NOT AT END
000530  010567  000002G                   MOV     R5,$TMLST+2       ; ELSE MARK NEW END
000534  010514                   55$:     MOV     R5,(R4)           ; LINK IN PREDECESSOR
000536  000207                   60$:     RETURN
                                 ;
                                 ; ERRORS
                                 ;
000540                           65$:     ERROR$  #ERRNSD           ; NO SUCH DTE
000550  005704                            TST     R4                ; WAS THE GROUP JUST CREATED?
000552  001026                            BNE     80$               ; BR IF ALREADY EXISTED
000554  012700  000000G                   MOV     #$XXAVL,R0        ; ELSE POINT TO FREE LIST
000560  012701  000000G                   MOV     #C$ULEN,R1        ; SIZE OF TEMPLATE
000564  010502                            MOV     R5,R2             ; ADDRESS OF TEMPLATE
000566  004767  000000G                   CALL    $RLBF             ; RETURN TEMPLATE TO FREE LIST
000572  000416                            BR      80$               ; RETURN FAILURE
000574                           70$:     ERROR$  #ERRDTE           ; MISSING DTE ID
000604  000411                            BR      80$
000606                           72$:     ERROR$  #ERRNUM           ; MISSING GROUP NUMBER
000616  000404                            BR      80$
000620                           75$:     ERROR$  #ERRALL           ; ALLOCATION FAILURE
```

```
                                               .SBTTL  SETLIN - SET LINE ID IN DTE TEMPLATE
                                ;+
                                ; ** SETLIN - SET LINE ID IN DTE TEMPLATE
                                ;
                                ; INPUTS         R5 = ADDRESS OF CURRENT DTE TEMPLATE
                                ;                $DVNAM = PARSED LINE ID
                                ;                $FLAGS = PARSED LINE ID ATTRIBUTES
                                ;
                                ; OUTPUT         THE PARSED LINE ID IS LOADED INTO THE
                                ;                DTE TEMPLATE. FLAGS IN THE STATUS WORD
                                ;                INDICATE PRESENCE OR ABSENCE OF UNIT
                                ;                AND TRIBUTARY NUMBERS.
                                ;
                                ; REGISTERS      R0 AND R1 ARE CLOBBERED.
                                ;-

000606                          SETLIN:
000606  010500                          MOV     R5,R0           ; COPY TEMPLATE POINTER
000610  062700  000000G                 ADD     #D$TLIN,R0      ; POINT PAST LINE ID
000614  012701  000000G                 MOV     #$DVNAM,R1      ; POINT TO PARSED LINE ID
        000004                          .REPT   4
                                        MOV     (R1)+,(R0)+     ; MOVE LINE ID
                                        .ENDR
000630  042765  000000C 000000G         BIC     #CS.UNI!CS.TRB,C.STS(R5) ; CLEAR LINE FLAGS
000636  032767  000000G 000000G         BIT     #FL.MUX,$FLAGS  ; MUX DEVICE ?
000644  001403                          BEQ     10$             ; IF EQ, NO
000646  052765  000000G 000000G         BIS     #CS.UNI,C.STS(R5); ELSE THERE IS A UNIT NUMBER
000654  032767  000000C 000000G 10$:    BIT     #FL.TRI!FL.TRB,$FLAGS ; MULTIPOINT LINE ?
000662  001403                          BEQ     20$             ; IF EQ, NOPE
000664  052765  000000G 000000G         BIS     #CS.TRB,C.STS(R5) ; ELSE THERE IS A TRIB NUMBER
000672  000207                  20$:    RETURN
```

```
                                   .SBTTL  LOCAL SYMBOL DEFINITIONS

                          ;****
                          ; LOCAL SYMBOL DEFINITIONS
                          ;****

          074000                   L.CUST = 480.*100          ; BEGINNING OF CUSTOMER SPECIFIC EVENTS

                          ;
                          ; DEFINE EVENT TEMPLATE
                          ;
000040                             .ASECT
          000000                   .=0
000000                             .BLKW    1                 ; C.LNK
000002                             .BLKW    1                 ; C.STS
          000004                   TMPLOC=.
000004                    C.CLS:   .BLKW    1                 ; EVENT CLASS
000006                    C.EVT:   .BLKW    4                 ; EVENT TYPE MASKS
000016                    C.SNK:   .BLKW    1                 ; LOGGING SINK TYPE
          000020          C.LEN=.
                          ;
                          ; DEFINE LOGGING STATE TEMPLATE
                          ;
          000004                   .=TMPLOC
000004                    C.STA:   .BLKB    1                 ; LOGGING STATE
          000005          C.LEN1=.
000160                             .PSECT
```

```
                                        .SBTTL  MACRO DEFINITIONS

                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MCALL  CALLR
                                ;
                                ; LOCAL DEFINITIONS
                                ;
        000177                  DEFPRI  = 127.          ; DEFAULT LIST INSERTION PRIORITY FOR DESTINATIONS

```

```
    498                                                 .SBTTL  $SEXAC - DEFINE MODULE X25-ACCESS
    499                                         ;+
    500                                         ; *** - $SEXPR - DEFINE MODULE X25-PROTOCOL
    501                                         ;
    502                                         ; INPUT:
    503                                         ;       $OPRND = REQUEST CODE
    504                                         ;
    505                                         ; OUTPUT:
    506                                         ;       THE REMOTE DESTINATION PARAMETERS ARE CHANGED.
    507                                         ;       IF THE REMOTE DESTINATION DOES NOT ALREADY EXIST
    508                                         ;       IT IS CREATED.
    509                                         ;
    510                                         ;-
    511
    512 002472                                  $SEXAC::
    513 002472  004767  000272                          CALL    TSTPSI                  ; MAKE SURE PSI IN SYSTEM
    514 002476  012702  000000G                         MOV     #$PSIPT,R2              ; POINT TO LISTHEAD
    515 002502  010205                          10$:    MOV     R2,R5                   ; SAVE PREVIOUS ADDRESS
    516 002504  011202                                  MOV     (R2),R2                 ; NEXT ADDRESS
    517 002506  001427                                  BEQ     30$                     ; BR IF END OF LIST
    518 002510  126227  000000G 000000G                 CMPB    C.STS(R2),#CS.PSN       ; CHECK FOR A GOOD PLACE TO LINK
    519 002516  002771                                  BLT     10$                     ; IF LT, BAD PLACE
    520 002520  126227  000000G 000000G                 CMPB    C.STS(R2),#CS.RDT       ; CHECK AGAIN
    521 002526  003365                                  BGT     10$                     ; IF GT, NO A GOOD PLACE TO LINK
    522 002530  001014                                  BNE     25$                     ; IF NE, WRONG TYPE BUT OK TO LINK AFTER
    523 002532  032767  000000G 000000G 20$:    BIT     #XA.KND,$OPRND          ; CHECK FOR KNOWN DTES
    524 002540  001056                                  BNE     50$                     ; BR IF WILDCARD
    525 002542  026762  000000G 000000G                 CMP     $DST,R$DNAM(R2)         ; CHECK FIRST PART OF NAME
    526 002550  001004                                  BNE     25$                     ; IF NE, WRONG DTE, BUT GOOD PLACE TO LINK
    527 002552  026762  000002G 000002G                 CMP     $DST+2,R$DNAM+2(R2) ; COMPARE SECOND WORD
    528 002560  001446                                  BEQ     50$                     ; IF EQ, IT MATCHES
    529 002562  010203                          25$:    MOV     R2,R3                   ; ELSE ITS A GOOD PLACE TO PUT A NEW ONE
    530 002564  000746                                  BR      10$                     ; KEEP ON LOOKING
    531                                         ;
    532                                         ; ALLOCATE A NEW REMOTE DESTINATION DESCRIPTOR
    533                                         ;
    534 002566  032767  000000G 000000G 30$:    BIT     #XA.KND,$OPRND          ; KNOWN DESTINATIONS
    535 002574  001062                                  BNE     65$                     ; IF NE, DONE
    536 002576  032767  000000G 000000G                 BIT     #XA.NUM,$OPRND          ; UPDATE DTE ADDRESS?
    537 002604  001464                                  BEQ     75$                     ; IF EQ,NO - ERROR ON CREATE
    538 002606  012700  000000G                         MOV     #$XXAVL,R0              ; POINT TO FREE LIST
    539 002612  012701  000000G                         MOV     #R$DLEN,R1              ; LENGTH TO REQUEST
    540 002616  004767  000170                          CALL    RQCB                    ; REQUEST CORE BLOCK
    541 002622  103450                                  BCS     70$                     ; BR IF ALLOCATION FAILURE
    542 002624  062767  000000G 000000G                 ADD     #R$LENX,$BYTMN          ; UPDATE MINIMUM BYTE ALLOCATION
    543 002632  062767  000000G 000000G                 ADD     #R$LENX,$BYTXT          ; UPDATE ACTUAL BYTE ALLOCATION
    544 002640  010002                                  MOV     R0,R2                   ; COPY POINTER
    545 002642  012762  000000G 000000G                 MOV     #CS.RDT,C.STS(R2)       ; INITIALIZE TEMPLATE
    546 002650  016762  000000G 000000G                 MOV     $DST,R$DNAM(R2)         ; GOTTA HAVE A NAME
    547 002656  016762  000002G 000002G                 MOV     $DST+2,R$DNAM+2(R2)     ; ...
    548 002664  011312                                  MOV     (R3),(R2)               ; LINK IN NEW SUCCESSOR
    549 002666  001002                                  BNE     40$                     ; BR UNLESS NEW CLEANUP
    550 002670  010267  000002G                         MOV     R2,$TMLST+2             ; UPDATE LISTHEAD
    551 002674  010213                          40$:    MOV     R2,(R3)                 ; LINK IN PREDECESSOR
    552 002676  032767  000000G 000000G 50$:    BIT     #XA.NUM,$OPRND          ; UPDATE DTE ADDRESS?
    553 002704  001412                                  BEQ     60$                     ; IF EQ, NO REASON TO BE HERE BUT NO ERROR
    554 002706  012700  000000G                         MOV     #$DTE,R0                ; POINT TO PARSED DTE ADDRESS
    112                                              ;
    113                                              ; DEFINE ROUTING TEMPLATE
    114                                              ;
    115                000000                                 .=0
    116 000000                                                .BLKW   1               ; C.LNK
    117 000002                                                .BLKW   1               ; C.STS
    118 000004                                       C.ADD:   .BLKW   1               ; MAXIMUM NODE ADDRESS
    119 000006                                       C.CST:   .BLKW   1               ; MAXIMUM COST ALLOWED
    120 000010                                       C.HOP:   .BLKW   1               ; MAXIMUM NUMBER OF HOPS ALLOWED
    121 000034                                                .PSECT
    122
    123                                              ;
    124                                              ; DEFINE REMOTE TEMPLATE
    125                                              ;
    126 000012                                                .ASECT
    127                000000                                 .=0
    128 000000                                                .BLKW   1               ; C.LNK
    129 000002                                                .BLKW   1               ; C.STS
    130 000004                                                .BLKW   2               ; REMOTE NODE NAME
    131 000010                                       C.RADD:  .BLKW   1               ; REMOTE NODE ADDRESS
    132 000034                                                .PSECT
    133
    134
```

```

                                                 .MACRO  ERROR$  TEXT
                                        ;
                                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                        ; MESSAGE STRING.
                                        ;
                                                 .IF DIF <TEXT><R0>
                                                 MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                                 .ENDC
                                                 CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                                 .ENDM   ERROR$

                                                 .MACRO  EPRINT  TEXT
                                        ;
                                        ; PRINT TEXT ON ERROR LUN
                                        ;
                                                 .IF DIF <TEXT><R0>
                                                 MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                                 .ENDC
                                                 CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                                 .ENDM   EPRINT

                                        ;****
                                        ; LOCAL DATA
                                        ;****

                                                 .NLIST  BEX
                                                 .ENABL  LC
               000002                            EX$ERR = 2
      000000      002      103      106 ERRDEC:  .ASCIZ  <EX$ERR>/CFE -- No DEC$DF macro/
      000030      002      103      106 ERRFEA:  .ASCIZ  <EX$ERR>/CFE -- No NS: features macro/
      000066      002      103      106 ERRSUP:  .ASCIZ  <EX$ERR>/CFE -- Access Verification not supported/
                                                 .EVEN
```

```
PRSMOD - PARSE ACTION ROUTINES  MACRO V05.03b Saturday 29-Jun-85 05:19  Page 27-1
Symbol table

CHNERR  000626R         $CHNND= ****** GX       $NNAM = ****** GX       .MOCRN  000636RG        .MOMXR  000350RG
ERRCAL= ****** GX       $CLM  = ****** GX       $NUM  = ****** GX       .MOCUG  000000RG        .MONNA  001252RG
ERRCHN= ****** GX       $CLT  = ****** GX       $OBJ  = ****** GX       .MODFD  000134RG        .MONUM  000522RG
ERRILN= ****** GX       $CLV  = ****** GX       $PRI  = ****** GX       .MODFW  000224RG        .MOOBJ  001176RG
ILLCAL  001040R         $CUG  = ****** GX       $RET  = ****** GX       .MODST  000660RG        .MOOBN  001152RG
ILLNUM  000506R         $DFD  = ****** GX       $RST  = ****** GX       .MODS1  001074RG        .MOPRI  001050RG
ILLNU2  001136R         $DFW  = ****** GX       $SUB1 = ****** GX       .MODS2  001110RG        .MORET  000324RG
REJECT  000514R         $DST  = ****** GX       $SUB2 = ****** GX       .MOINC  000552RG        .MORST  000374RG
REJEC2  001144R         $ERROR= ****** GX       .MOCAT  000054RG        .MOMCI  000450RG        .PNUMB= ****** GX
$BSZ  = ****** GX       $MCI  = ****** GX       .MOCLM  000736RG        .MOMCL  000254RG        .PNUMH= ****** GX
$CALT = ****** GX       $MRE  = ****** GX       .MOCLT  000104RG        .MOMC1  000420RG        .PSTCN= ****** GX
$CAT5 = ****** GX       $MXC  = ****** GX       .MOCLV  000744RG        .MOMDA  000170RG        .PSTPT= ****** GX
$CHNLS= ****** GX       $MXR  = ****** GX       .MOCNN  000560RG        .MOMRE  000300RG

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001326    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 76 Words  ( 1 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:13.26
SY:PRSMOD.V2,[132,134]PRSMOD/CR/-SP=SY:[1,1]RSXMAC.SML/ML,[130,110]NETLIB/ML,[132,10]PRSMOD
```

```
**FILE**ID**PRSPRO
```

```
PPPPPPPP     RRRRRRRR       SSSSSSSS   PPPPPPPP     RRRRRRRR       000000
PPPPPPPP     RRRRRRRR       SSSSSSSS   PPPPPPPP     RRRRRRRR       000000
PP      PP   RR      RR   SS           PP      PP   RR      RR   00      00
PP      PP   RR      RR   SS           PP      PP   RR      RR   00      00
PP      PP   RR      RR   SS           PP      PP   RR      RR   00      00
PP      PP   RR      RR   SS           PP      PP   RR      RR   00      00
PPPPPPPP     RRRRRRRR       SSSSSS     PPPPPPPP     RRRRRRRR     00      00
PPPPPPPP     RRRRRRRR       SSSSSS     PPPPPPPP     RRRRRRRR     00      00
PP           RR  RR             SS   PP           RR  RR       00      00
PP           RR  RR             SS   PP           RR  RR       00      00
PP           RR    RR           SS   PP           RR    RR     00      00       ....
PP           RR    RR           SS   PP           RR    RR     00      00       ....
PP           RR      RR   SSSSSSSS     PP           RR      RR     000000         ....
PP           RR      RR   SSSSSSSS     PP           RR      RR     000000         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                          .SBTTL  .RNHO1  - PARSE REMOTE NODE HOST
                                          .SBTTL  .RNHOS  - PARSE REMOTE NODE HOST
                                  ;+
                                  ; INPUTS:       $NODAD  = NODE ADDRESS AS PARSED
                                  ;
                                  ; OUTPUT:       $RNHOS  = HOST NODE ADDRESS
                                  ;-

000156                            .RNHO1::
000156  016767  000000G 000000G           MOV     $NODAD,$RNHOS   ; SAVE REMOTE NODE ADDRESS TEMPORARILY
000164  000207                            RETURN

000166                            .RNHO2::
000166  062716  000002                    ADD     #2,(SP)         ; RESTORE SANITY AFTER FAILED PARSE
000172                            .RNHOS::
000172  016700  000000G                   MOV     $RNHOS,R0       ; RECOVER SAVED REMOTE ADDRESS
000176  016767  000000G 000000G           MOV     $NODAD,$RNHOS   ; SAVE PARSED HOST ADDRESS
000204  010067  000000G                   MOV     R0,$NODAD       ; RESTORE REMOTE ADDRESS
000210  000207                            RETURN


                                          .SBTTL  .RNSDV - PARSE SERVICE DEVICE
                                  ;+
                                  ; **-.RNSDV-PARSE SERVICE DEVICF
                                  ;
                                  ; INPUTS:       TPARS VARIABLES.
                                  ;
                                  ; OUTPUT:       $RNSDV = RAD50 SERVICE DEVICE.
                                  ;-
000212                            .RNSDV::
000212  016700  000000G                   MOV     .PSTPT,R0       ; POINT TO IT
000216  004767  000000G                   CALL    $CAT5           ; CONVERT IT
000222  010167  000000G                   MOV     R1,$RNSDV       ; SAVE IT
000226  000207                            RETURN


                                          .SBTTL  .RNDPA - PARSE DUMP ADDRESS
                                  ;+
                                  ; **-.RNDPA-PARSE DUMP ADDRESS
                                  ;
                                  ; INPUTS:       TPARS VARIABLES.
                                  ;
                                  ; OUTPUT:       $RNDMP = 32 BIT DUMP ADDRESS
                                  ;-
000230                            .RNDPA::
000230  016767  000000G 000000G           MOV     .PNUMB,$RNDPA
000236  016767  000000G 000002G           MOV     .PNUMH,$RNDPA+2
000244  000207                            RETURN


                                          .SBTTL  .RNDCT - PARSE DUMP COUNT
                                  ;+
                                  ; **-.RNDCT-PARSE DUMP COUNT
                                  ;
                                  ; INPUTS:       .PNUMB = DUMP COUNT IN 64. BYTE UNITS
                                  ;
                                  ; OUTPUT:       $RNDCP = DUMP COUNT IN 1024. BYTE UNITS
```

```
        .TITLE  PRSSYS - PARSE SYSTEM ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - PARSE COMMANDS ASSOCIATED WITH THE SYSTEM PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 4.01  7-Dec-84        LAA
;       Changed byte extension max from 8192. to 65472.
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                                                                                J 5
**FILE**ID**PURMOD




PPPPPPPP     UU      UU   RRRRRRRR     MM      MM     OOOOOO     DDDDDDDD
PPPPPPPP     UU      UU   RRRRRRRR     MM      MM     OOOOOO     DDDDDDDD
PP      PP   UU      UU   RR      RR   MMMM  MMMM   OO      OO   DD      DD
PP      PP   UU      UU   RR      RR   MMMM  MMMM   OO      OO   DD      DD
PP      PP   UU      UU   RR      RR   MM  MM  MM   OO      OO   DD      DD
PP      PP   UU      UU   RR      RR   MM  MM  MM   OO      OO   DD      DD
PPPPPPPP     UU      UU   RRRRRRRR     MM      MM   OO      OO   DD      DD
PPPPPPPP     UU      UU   RRRRRRRR     MM      MM   OO      OO   DD      DD
PP           UU      UU   RR  RR       MM      MM   OO      OO   DD      DD
PP           UU      UU   RR  RR       MM      MM   OO      OO   DD      DD
PP           UU      UU   RR    RR     MM      MM   OO      OO   DD      DD       ....
PP           UU      UU   RR    RR     MM      MM   OO      OO   DD      DD       ....
PP           UUUUUUUUUU   RR      RR   MM      MM     OOOOOO     DDDDDDDD         ....
PP           UUUUUUUUUU   RR      RR   MM      MM     OOOOOO     DDDDDDDD         ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                .SBTTL  X29CTM - PURGE X29-SERVER COUNTER TIMER
                                        ;+
                                        ; ** X29CTM - PURGE X29-SERVER COUNTER TIMER
                                        ;
                                        ; INPUTS        NONE.
                                        ;
                                        ; OUTPUT        THE X29-SERVER COUNTER TIMER IS PURGED.
                                        ;-

 001650                                 X29CTM:
 001650  012700  000000G                        MOV     #$TMLST,R0      ; POINT TO LISTHEAD
 001654  011000                         10$:    MOV     (R0),R0         ; NEXT
 001656  001407                                 BEQ     101$            ; IF EQ, NO X29 SUPPORT
 001660  122760  000000G 000000G                CMPB    #CS.X29,C.STS(R0) ; CHECK FOR X29 DATABASE
 001666  001372                                 BNE     10$             ; KEEP ON LOOKING
 001670  005060  000000G                        CLR     X$9CT(R0)       ; PURGE THE COUNTER TIMER
 001674  000207                                 RETURN
                                        ;
                                        ; ERRORS
                                        ;
 001676                                 101$:   ERROR$  #ERRX29         ; NO X29 SUPPORT
 001706  000207                                 RETURN

```

```

000016                      .ASECT
          000000            . = 0
000000              F.LNK:  .BLKW   1               ; Link word
000002              F.CNT:  .BLKW   1               ; Use count
000004                      .BLKB   1               ; Filespec string length
000005              F.DEV:  .BLKB   MXFSP           ; Filespec string (ASCII)
          000043    F.LEN1  = .
000000                      .PSECT
                    ;
                    ; LOCAL DEFINITIONS
                    ;
          000012    REMLEN = 12                     ; LENGTH OF REMOTE TEMPLATE
000000    000000    SUCC:   .WORD   0               ; SUCCESS FLAG
000002              SERPT:  .BLKW   1               ; ADDRESS OF SERVICE TEMPLATE
000004              FILPT:  .BLKW   1               ; ADDRESS OF FILES TEMPLATE
```

Table of contents

5- 158 $SEBUF - SETUP BUFFER TEMPLATE

```
                                                .SBTTL  $SECNT - SET CONTROLLER PARAMETERS
                                        ;+
                                        ; *** - $SECNT - SET CONTROLLER PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       THE CONTROLLER TEMPLATE IS UPDATED
                                        ;
                                        ;-

000040                                  $SECNT::
000040  012705  000000G                         MOV     #$TMLST,R5      ; GET START OF TEMPLATES LIST
000044  005001                                  CLR     R1              ; SUCCESS INDICATOR
                                        ;
                                        ; LOOK FOR THE DDM PROCESS NAME
                                        ;
000046  011505                          10$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
000050  001517                                  BEQ     65$             ; IF EQ, END OF TEMPLATES
000052  122765  000000G 000000G                 CMPB    #CS.DDM,C.STS(R5) ; IS THIS A DDM TEMPLATE ?
000060  001372                                  BNE     10$             ; IF NE, NO - KEEP LOOKING
000062  032767  000000G 000000G 15$:    BIT     #WC.NAM,$WILD   ; ALL DEVICES ?
000070  001004                                  BNE     20$             ; IF NE, YES .. FIND CONTROLLER
000072  026765  000000G 000000G                 CMP     $DVNAM,C.PRM(R5) ; IS THIS THE RIGHT DEVICE NAME ?
000100  001362                                  BNE     10$             ; IF NE, NO .. KEEP LOOKING
000102  010500                          20$:    MOV     R5,R0           ; SAVE DDM TEMPLATE ADDRESS
                                        ;
                                        ; LOOK FOR THE CONTROLLER
                                        ;
000104  011505                          25$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
000106  001500                                  BEQ     65$             ; IF EQ, END OF TEMPLATES
000110  122765  000000G 000000G                 CMPB    #CS.DDM,C.STS(R5) ; IS THIS THE NEXT DDM ?
000116  001761                                  BEQ     15$             ; BR IF YES
000120  122765  000000G 000000G                 CMPB    #CS.CNT,C.STS(R5) ; IS THIS A CONTROLLER TEMPLATE ?
000126  001366                                  BNE     25$             ; IF NE, NO .. KEEP LOOKING
000130  032767  000000C 000000G                 BIT     #WC.NAM!WC.CNT,$WILD    ; WILDCARD DEVICES/CONTROLLERS ?
000136  001004                                  BNE     30$             ; IF NE, YES
000140  026765  000000G 000004                  CMP     $CNTNM,C.CNT(R5) ; IS THIS THE RIGHT CONTROLLER ?
000146  001356                                  BNE     25$             ; IF NE, NO .. KEEP LOOKING
                                        ;
                                        ; UPDATE THE CONTROLLER TEMPLATE
                                        ;
000150  052765  000000G 000000G 30$:    BIS     #CS.MOD,C.STS(R5) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000156  005201                                  INC     R1              ; SHOW WE FOUND ONE
000160  032767  000000G 000000G                 BIT     #OL.URM,$OPRND  ; IS THE UNIBUS RUN MASK PRESENT ?
000166  001403                                  BEQ     35$             ; IF EQ, NO
000170  016765  000000G 000014                  MOV     $URM,C.URM(R5)  ; ELSE, UPDATE THE UNIBUS RUN MASK
000176  032767  000000G 000000G 35$:    BIT     #OL.VEC,$OPRND  ; IS THE VECTOR ADDRESS PRESENT
000204  001403                                  BEQ     40$             ; IF EQ, NO
000206  016765  000000G 000006                  MOV     $VEC,C.VEC(R5)  ; ELSE, UPDATE THE VECTOR ADDRESS
000214  032767  000000G 000000G 40$:    BIT     #OL.CSR,$OPRND  ; IS THE CSR ADDRESS PRESENT ?
000222  001403                                  BEQ     45$             ; IF EQ, NO
000224  016765  000000G 000010                  MOV     $CSR,C.CSR(R5)  ; ELSE, UPDATE THE CSR ADDRESS
000232  032767  000000G 000000G 45$:    BIT     #OL.PRI,$OPRND  ; IS THE PRIORITY PRESENT
000240  001422                                  BEQ     60$             ; IF EQ, NO .. LOOK FOR MORE WORK
```

```
    000630  000261                  80$:    SEC
    000632  000207                          RETURN

            000001                          .END
```

```
                                                .SBTTL  SETCHN - CREATE CHANNEL LIST
                                          ;+
                                          ; ** SETCHN - CREATE CHANNEL LIST
                                          ;
                                          ; INPUTS          R5 = ADDRESS OF CURRENT DTE TEMPLATE
                                          ;                 $CHNLS = PARSED CHANNEL LIST
                                          ;
                                          ; OUTPUT          THE CURRENT CHANNEL LIST IS PURGED, AND THE
                                          ;                 PARSED CHANNEL LIST IS CREATED.
                                          ;
                                          ;                 CARRY SET IF ALLOCATION FAILURE.
                                          ;
                                          ; REGISTERS       R5 IS PRESERVED.
                                          ;-

000674                                    SETCHN:
                                          ;
                                          ; PURGE THE CURRENT TEMPLATE LIST
                                          ;
000674  010546                                  MOV     R5,-(SP)          ; SAVE R5
000676  010502                            10$:  MOV     R5,R2             ; COPY TEMPLATE POINTER
000700  011202                                  MOV     (R2),R2           ; NEXT TEMPLATE
000702  001417                                  BEQ     30$               ; IF EQ, END OF LIST
000704  122762  000000G 000000G                 CMPB    #CS.CHN,C.STS(R2) ; CHECK TEMPLATE TYPE
000712  001013                                  BNE     30$               ; IF NE, END OF CHANNEL DEFS
000714  011215                                  MOV     (R2),(R5)         ; CLOSE UP LIST
000716  001002                                  BNE     20$               ; IF NE, NOT LAST IN LIST
000720  010567  000002G                         MOV     R5,$TMLST+2       ; SET NEW LAST ENTRY
000724  012701  000000G                   20$:  MOV     #C$HLEN,R1        ; GET LENGTH OF CHANNEL TEMPLATE
000730  016700  000000G                         MOV     $XXAVL,R0         ; POINT AT END OF TASK LISTHEAD
000734  004767  000000G                         CALL    $RLBF             ; RELEASE CORE BLOCK
000740  000756                                  BR      10$               ; CONTINUE
                                          ;
                                          ; BUILD THE NEW CHANNEL LIST
                                          ;
000742  012704  000000G                   30$:  MOV     #$CHNLS,R4        ; POINT TO PARSED CHANNEL LISTHEAD
000746  012403                                  MOV     (R4)+,R3          ; GET COUNT OF CHANNELS
000750  006203                                  ASR     R3                ; CONVERT WORD OFFSET TO COUNT
000752  005303                            40$:  DEC     R3                ; REDUCE SAVED CHANNEL COUNT
000754  000241                                  CLC                       ; ASSUME SUCCESS
000756  100435                                  BMI     60$               ; IF MI, NO MORE
000760  012700  000000G                         MOV     #$XXAVL,R0        ; GET ADDRESS OF END OF TASK LISTHEAD
000764  012701  000000G                         MOV     #C$HLEN,R1        ; LENGTH OF CHANNEL TEMPLATE
000770  004767  000000G                         CALL    $RQBF             ; TRY TO ALLOCATE BLOCK
000774  103426                                  BCS     60$               ; IF CS, ALLOCATION FAILURE
000776  011510                                  MOV     (R5),(R0)         ; POINT TO NEW SUCCESSOR
001000  001002                                  BNE     50$               ; IF NE, NOT LAST
001002  010067  000002G                         MOV     R0,$TMLST+2       ; UPDATE END POINTER
001006  010015                            50$:  MOV     R0,(R5)           ; LINK INTO LIST
001010  010005                                  MOV     R0,R5             ; WHERE TO LINK NEXT ONE
001012  012760  000000G 000000G                 MOV     #CS.CHN,C.STS(R0) ; ITS A CHANNEL BLOCK
001020  012460  000000G                         MOV     (R4)+,C$HLO(R0)   ; SET LOW END OF RANGE
001024  100352                                  BPL     40$               ; BR IF NO HIGH CHANNEL
001026  042760  100000  000000G                 BIC     #100000,C$HLO(R0); CLEAR FLAG BIT
001034  052760  000000G 000000G                 BIS     #CH$HI,C.STS(R0)  ; REMEMBER LOW CHANNEL SPECIFIED
001042  012460  000000G                         MOV     (R4)+,C$HHI(R0)   ; SAVE LOW END OF RANGE
001046  005303                                  DEC     R3                ; ACCOUNT FOR EXTRA CHANNEL
```

```
                                                .SBTTL  $SELOG - SET LOGGING PARAMETERS
                                        ;+
                                        ; *** - $SELOG - SET LOGGING PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       THE EVENT AND/OR LOGGING STATE TEMPLATES ARE MODIFIED
                                        ;
                                        ;-
000160                                  $SELOG::
000160  012705  000007                          MOV     #FF.MON!FF.CON!FF.FIL,R5 ; ASSUME KNOWN LOGGING
000164  032767  000000G 000000G                 BIT     #WC.LOG,$WILD           ; KNOWN LOGGING SPECIFIED?
000172  001016                                  BNE     10$                     ; BR IF YES
000174  012705  000004                          MOV     #FF.MON,R5              ; ASSUME LOGGING MONITOR
000200  032767  000000G 000000G                 BIT     #OM.MON,$OPRND          ; LOGGING MONITOR SPECIFIED?
000206  001010                                  BNE     10$                     ; BR IF YES
000210  012705  000001                          MOV     #FF.CON,R5              ; ASSUME LOGGING CONSOLE
000214  032767  000000G 000000G                 BIT     #OM.CON,$OPRND          ; LOGGING CONSOLE SPECIFIED?
000222  001002                                  BNE     10$                     ; BR IF YES
000224  012705  000002                          MOV     #FF.FIL,R5              ; ELSE MUST BE LOGGING FILE
000230  032767  000000G 000000G 10$:            BIT     #OM.STA,$OPRND          ; LOGGING STATE SPECIFIED?
000236  001432                                  BEQ     50$                     ; BR IF NO
                                        ;
                                        ; PROCESS LOGGING STATE
                                        ;
000240  012700  000000G                         MOV     #$TMLST,R0              ; GET START OF TEMPLATES
000244  011000                          20$:    MOV     (R0),R0                 ; GET NEXT TEMPLATE
000246  001006                                  BNE     30$                     ; BR IF FOUND ONE
000250                                          ERROR$  #ERRLOG                 ; LOGGING STATE TEMPLATE NOT FOUND
000260  000167  000442                          JMP     150$                    ; AND LEAVE
000264  122760  000000G 000000G 30$:            CMPB    #CS.LOG,C.STS(R0)       ; IS THIS A LOGGING TEMPLATE
000272  001364                                  BNE     20$                     ; BR IF NO -- KEEP LOOKING
000274  052760  000000G 000000G                 BIS     #CS.MOD,C.STS(R0)       ; INDICATE LOGGING STATE MODIFIED
000302  032767  000000G 000000G                 BIT     #SW.LON,$SWICH          ; SET LOGGING STATE ON?
000310  001003                                  BNE     40$                     ; BR IF YES
000312  140560  000004                          BICB    R5,C.STA(R0)            ; SET LOGGING STATE OFF
000316  000402                                  BR      50$                     ; CONTINUE
000320  150560  000004                  40$:    BISB    R5,C.STA(R0)            ; SET LOGGING STATE ON
                                        ;
                                        ; PROCESS EVENT DEFINITION
                                        ;
000324  032767  000000G 000000G 50$:            BIT     #OM.EVE,$OPRND          ; EVENTS SPECIFIED?
000332  001575                                  BEQ     150$                    ; BR IF NO
000334  032767  000000G 000000G                 BIT     #WC.EVE,$WILD           ; ALL EVENTS SPECIFIED?
000342  001412                                  BEQ     60$                     ; BR IF NO
000344  012704  000000G                         MOV     #$KNEVT,R4              ; ELSE POINT TO KNOWN EVENT TABLE
000350  021467  000000G                 55$:    CMP     (R4),$CLASS             ; MATCH ON EVENT CLASS
000354  001413                                  BEQ     70$                     ; IF EQ, GOT IT
000356  062704  000012                          ADD     #12,R4                  ; POINT TO NEXT ENTRY
000362  005714                                  TST     (R4)                    ; END OF TABLE ?
000364  100371                                  BPL     55$                     ; IF PL, MORE
000366  000434                                  BR      103$                    ; ELSE UNKNOWN EVENT CLASS
000370  032767  000000G 000000G 60$:            BIT     #OM.KEV,$OPRND          ; KNOWN EVENTS SPECIFIED?
000376  001415                                  BEQ     80$                     ; BR IF NO
```

```
     82                                                         .SBTTL  LOCAL DATA
     83                                                         .NLIST  BEX
     84
     85                                                 ;****
     86                                                 ; LOCAL DATA
     87                                                 ;****
     88
     89                                                 ;
     90                                                 ; ERROR TEXT
     91                                                 ;
     92                                                         .NLIST  BEX
     93                 000002                                  EX$ERR = 2
     94 000000             002        103        106    ERRPSI: .ASCIZ  <EX$ERR>/CFE -- No PSI support in system/
     95 000041             002        103        106    ERRDNM: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: OBJECT/
     96 000103             002        103        106    ERRCON: .ASCIZ  <EX$ERR>/CFE -- Invalid destination sequence/
     97 000150             002        103        106    ERRDST: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: DESTINATION/
     98 000217             002        103        106    ERROBJ: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: OBJECT/
     99 000261             002        103        106    ERRCLM: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: CALL MASK/
    100 000326             002        103        106    ERRCLV: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: CALL VALUE/
    101 000374             002        103        106    ERRNUM: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: NUMBER/
    102 000436             002        103        106    ERRXSV: .ASCIZ  <EX$ERR>/CFE -- Missing LLC$DF for NW/
    103 000474             002        103        106    ERRX29: .ASCIZ  <EX$ERR>/CFE -- No X.29 support in configuration/
    104 000545             002        103        106    ERRLEN: .ASCIZ  <EX$ERR>/CFE -- Call mask and call value are different lengths/
    105                                                         .EVEN
    106                                                         .LIST BEX
    107
    555 002712  010201                             MOV     R2,R1              ; COPY TEMPLATE ADDRESS
    556 002714  062701  000000G                    ADD     #R$DADR,R1         ; POINT TO NAME FIELD
    557                 000005                     .REPT   5
    558                                            MOV     (R0)+,(R1)+        ; LOAD IT UP
    559                                            .ENDR
    560 002732  032767  000000G 000000G 60$:       BIT     #XA.KND,$OPRND     ; WILDCARD OPERATION?
    561 002740  001260                             BNE     10$                ; IF NE, CONTINUE
    562 002742  000207                  65$:       RETURN
    563                                 ;
    564                                 ; ERRORS
    565                                 ;
    566 002744                          70$:       ERROR$  #ERRALL            ; ALLOCATION FAILURE
    567 002754  000207                             RETURN
    568
    569 002756                          75$:       ERROR$  #ERRNUM            ; MISSING NUMBER ON CREATE
    570 002766  000207                             RETURN
```

```
                                        .SBTTL  $SENOD - SET NODE PARAMETERS
                                ;+
                                ; *** - $SENOD - SET NODE PARAMETERS
                                ;
                                ; INPUT:
                                ;       $OPRND = OPERANDS PRESENT
                                ;
                                ; OUTPUT:
                                ;       THE NODE TEMPLATE IS MODIFIED
                                ;
                                ;-

000034                          $SENOD::
000034  012700  000000G                 MOV     #$TMLST,R0      ; GET START OF TEMPLATES
                                ;
                                ; SEARCH FOR THE NODE TEMPLATE
                                ;
000040  011000                  10$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
000042  001513                          BEQ     101$            ; IF EQ, NO NODE TEMPLATE
000044  122760  000000G 000000G         CMPB    #CS.NOD,C.STS(R0) ; IS THIS A NODE TEMPLATE ?
000052  001372                          BNE     10$             ; IF NE, NO - KEEP LOOKING
000054  052760  000000G 000000G         BIS     #CS.MOD,C.STS(R0) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000062  032767  000000G 000000G         BIT     #OE.NAM,$OPRND  ; IS THERE A NODE NAME ?
000070  001414                          BEQ     15$             ; IF EQ, NO
000072  016760  000000G 000004          MOV     $NODNA,C.NAM(R0) ; ELSE, STORE THE NEW NODE NAME
000100  016760  000002G 000006          MOV     $NODNA+2,C.NAM+2(R0) ; ...
000106  016767  000000G 000000G         MOV     $NODNA,$NTNAM   ; ..TWICE
000114  016767  000002G 000002G         MOV     $NODNA+2,$NTNAM+2 ; ...
000122  032767  000000G 000000G 15$:    BIT     #OE.ADD,$OPRND  ; IS THERE A NODE ADDRESS ?
000130  001421                          BEQ     20$             ; IF EQ, NO
000132  042760  001777  000052          BIC     #1023.,C.NUM(R0); PRESERVE OLD AREA
000140  016001  000052                  MOV     C.NUM(R0),R1    ; SAVE IT
000144  032767  176000  000000G         BIT     #^C1023.,$NODAD ; AREA SPECIFIED FOR NEW ?
000152  001402                          BEQ     18$             ; IF EQ, NO - USE OLD AREA
000154  005060  000052                  CLR     C.NUM(R0)       ; ELSE, CLEAR OUT AREA BITS
000160  056760  000000G 000052  18$:    BIS     $NODAD,C.NUM(R0); SET NEW ADDRESS AND POSSIBLY AREA
000166  016067  000052  000000G         MOV     C.NUM(R0),$NTADD ; ..TWICE
                                ;;;     MOV     $NODAD,R2       ; PUSH NEW ADDRESS
                                ;;;     BIC     #1777,R2        ; ISOLATE NEW AREA
                                ;;;     CMP     R1,R2           ; IS IT THE SAME AREA ?
                                ;;;     BEQ     20$             ; BR IF YES
                                ;;;     CALL    UPDREM          ; ELSE UPDATE REMOTE DEFINITIONS
000174  032767  000000G 000000G 20$:    BIT     #OE.ID,$OPRND   ; IS THERE A NODE ID ?
000202  001412                          BEQ     30$             ; IF EQ, NO
000204  010046                          MOV     R0,-(SP)        ; SAVE R0
000206  062700  000010                  ADD     #C.NID,R0       ; POINT R0 AT OUTPUT BUFFER
000212  012701  000000G                 MOV     #$ID,R1         ; GET ADDRESS OF INPUT BUFFER
000216  012102                          MOV     (R1)+,R2        ; AND LENGTH OF STRING TO MOVE
000220  010220                          MOV     R2,(R0)+        ; TWICE
000222  004767  000000G                 CALL    $MVASC          ; MOVE THE NODE ID INTO TEMPLATE
000226  012600                          MOV     (SP)+,R0        ; RESTORE R0
000230  032767  000000G 000000G 30$:    BIT     #OE.HOS,$OPRND  ; IS THERE A HOST NODE ADDRESS ?
000236  001414                          BEQ     40$             ; IF EQ, NO
000240  042760  001777  000054          BIC     #1023.,C.HOS(R0) ; PRESERVE AREA
000246  032767  176000  000000G         BIT     #^C1023.,$HOST  ; AREA IN HOST ?
000254  001402                          BEQ     35$             ; IF EQ, NO - USE PREVIOUS
000256  005060  000054                  CLR     C.HOS(R0)       ; ELSE, CLEAR AREA BITS
```

```
                                                .SBTTL  $SENSP - SET NSP PARAMETERS
                                        ;+
                                        ; *** - $SENSP - SET NSP PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       THE NSP TEMPLATE IS MODIFIED
                                        ;
                                        ;-

000140                                  $SENSP::
                                        ;
                                        ; SEARCH FOR THE NSP TEMPLATE
                                        ;
000140  010546                                  MOV     R5,-(SP)        ; SAVE REGISTER
000142  012705  000000G                         MOV     #$TMLST,R5      ; GET START OF TEMPLATES
000146  010504                                  MOV     R5,R4           ; AGAIN
000150  011505                          10$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
000152  001555                                  BEQ     110$            ; IF EQ, NO DEC$DF TEMPLATE
000154  122765  000000G 000000G                 CMPB    #CS.DEC,C.STS(R5) ; IS THIS THE DEC$DF TEMPLATE?
000162  001372                                  BNE     10$             ; IF NE, NO - KEEP LOOKING
000164  010504                                  MOV     R5,R4           ; GET ADDRESS OF DEC$DF TEMPLATE
                                        ;
                                        ; UPDATE THE DEC$DF TEMPLATES
                                        ;
000166  032767  000000G 000000G                 BIT     #OE.LNK,$OPRND  ; WAS THE LINK COUNT CHANGED ?
000174  001406                                  BEQ     20$             ; IF EQ, NO
000176  016764  000000G 000010                  MOV     $LINKS,C.LKS(R4) ; UPDATE THE NUMBER OF LINKS
000204  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED

000212  032767  000000G 000000G 20$:            BIT     #OE.NCO,$OPRND  ; WAS THE NUMBER OF NODE COUNTERS SPECIFIED ?
000220  001406                                  BEQ     30$             ; IF EQ, NO
000222  016764  000000G 000006                  MOV     $NCOU,C.NDC(R4) ; UPDATE THE NUMBER OF NODE COUNTERS
000230  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED

000236  032767  000000G 000000G 30$:            BIT     #E2.INA,$OPRN2  ; WAS INACTIVITY TIMER SPECIFIED?
000244  001406                                  BEQ     40$             ; BR IF NO
000246  116764  000000G 000014                  MOVB    $EINA,C.INAC(R4) ; UPDATE THE INACTIVITY TIMER
000254  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDCATE TEMPLATE WAS MODIFIED

000262  032767  000000G 000000G 40$:            BIT     #E2.INC,$OPRN2  ; WAS INCOMING TIMER SPECIFIED?
000270  001406                                  BEQ     50$             ; BR IF NO
000272  116764  000000G 000012                  MOVB    $EINC,C.INCT(R4) ; UPDATE THE INCOMING TIMER
000300  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDCATE TEMPLATE WAS MODIFIED

000306  032767  000000G 000000G 50$:            BIT     #E2.OUT,$OPRN2  ; WAS OUTGOING TIMER SPECIFIED?
000314  001406                                  BEQ     60$             ; BR IF NO
000316  116764  000000G 000013                  MOVB    $EOUT,C.OUTT(R4) ; UPDATE THE OUTGOING TIMER
000324  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDCATE TEMPLATE WAS MODIFIED

000332  032767  000000G 000000G 60$:            BIT     #E2.RET,$OPRN2  ; WAS RETRANSMIT FACTOR SPECIFIED?
000340  001406                                  BEQ     70$             ; BR IF NO
000342  116764  000000G 000015                  MOVB    $ERET,C.RETF(R4) ; UPDATE THE RETRANSMIT FACTOR
000350  052764  000000G 000000G                 BIS     #CS.MOD,C.STS(R4) ; INDCATE TEMPLATE WAS MODIFIED

```

K 1

```
PRSMOD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE              REFERENCES

CHNERR     000626 R          19-373    #19-377    20-396
ERRCAL   = ******   GX       22-467
ERRCHN   = ******   GX       19-377
ERRILN   = ******   GX       16-313     24-517
ILLCAL     001040 R          22-448    #22-467
ILLNUM     000506 R          5-99       6-117     7-130      7-133      7-135      8-152      8-155      8-157      9-173
                             9-175      9-177     10-192     10-194     11-210     11-212     12-228     12-230     13-246
                             13-248     14-264    14-266     15-283     15-285     15-287     16-303     16-307    #16-313
                             17-329     17-331    17-333     19-364     19-366     19-368     23-484     23-486
ILLNU2     001136 R          24-509     24-512   #24-517     25-533     25-535
REJECT     000514 R         #16-314     19-378    22-468
REJEC2     001144 R         #24-518     26-550    27-574
$BSZ     = ******   GX      *8-158
$CALT    = ******   GX      *5-97
$CAT5    = ******   GX       4-72       4-76      21-414     21-418     22-459     26-553     26-558     27-577     27-582
$CHNLS   = ******   GX      *18-348     19-369    20-393
$CHNND   = ******   GX       19-372
$CLM     = ******   GX       22-441
$CLT     = ******   GX      *6-115
$CLV     = ******   GX       22-445
$CUG     = ******   GX      *4-69      *4-73      *4-77
$DFD     = ******   GX      *7-136
$DFW     = ******   GX      *9-176
$DST     = ******   GX      *21-411    *21-415    *21-419
$ERROR   = ******   GX      *16-313    *19-377    *22-467    *24-517
$MCI     = ******   GX      *15-286    *16-310
$MRE     = ******   GX      *11-213
$MXC     = ******   GX      *10-195
$MXR     = ******   GX      *13-249
$NNAM    = ******   GX      *27-578    *27-579    *27-583
$NUM     = ******   GX      *17-332
$OBJ     = ******   GX      *25-536    *26-554    *26-555    *26-559
$PRI     = ******   GX      *23-487
$RET     = ******   GX      *12-231
$RST     = ******   GX      *14-267
$SUB1    = ******   GX       24-503
$SUB2    = ******   GX      *24-504     24-507
.MOCAT     000054 RG        #5-92
.MOCLM     000736 RG        #22-440
.MOCLT     000104 RG        #6-110
.MOCLV     000744 RG        #22-444
.MOCNN     000560 RG        #19-362
.MOCPN     000636 RG        #20-392
.MOCUG     000000 RG        #4-68
.MODFD     000134 RG        #7-128
.MODFW     000224 RG        #9-171
.MODST     000660 RG        #21-410
.MODS1     001074 RG        #24-502
.MODS2     001110 RG        #24-506
.MOINC     000552 RG        #18-347
.MOMCI     000450 RG        #16-301
.MOMCL     000254 RG        #10-190
```

L 1

Table of contents

```
     4-   55    .PRNAM - PROCESS NAME ACTION ROUTINE
     5-   86    .PRNLT - STORE NUMBER OF LINE TABLES SPECIFIED
     6-  106    .PRNCT - STORE NUMBER OF CONTROLLER TABLES SPECIFIED
```

```
                                        ;                               (ROUNDED UP TO THE NEAREST 1K BOUNDARY)
                                        ;-
000246                                  .RNDCT::
000246  016700  000000G                         MOV     .PNUMB,R0       ; GRAB IT
000252  001416                                  BEQ     ILLNUM          ; ZERO NOT ALLOWED
000254  005767  000000G                         TST     .PNUMH          ; GOTTA BE A 16 BIT VALUE
000260  001013                                  BNE     ILLNUM          ; ....
000262  062700  000017                          ADD     #1023./64.,R0   ; ROUND IT UP TO THE NEAREST 1 K
000266  042700  000017                          BIC     #1023./64.,R0   ; ...
000272                                          ASR$    4,R0            ; IN 1K UNITS
000302  010067  000000G                         MOV     R0,$RNDCP       ; SAVE IT
000306  000207                                  RETURN

000310  012767  000000G 000000G ILLNUM: MOV     #ERRILN,$ERROR  ; INVALID NUMBER

000316  062716  000002          REJ:    ADD     #2,(SP)         ; REJECT THE TRANSITION
000322  000207                                  RETURN




                                                .SBTTL  .RNSCI - STORE SERVICE CIRCUIT
                                        ;+
                                        ; *** .RNSCI - STORE SERVICE CIRCUIT
                                        ;
                                        ; INPUT:        .PSTPT  = ADDRESS OF STRING
                                        ;               .PSTCN  = STRING LENGTH
                                        ;
                                        ; OUTPUT:
                                        ;               $RNSCI  = SERVICE CIRCUIT (RAD50)
                                        ;
                                        ;-
000324                                  .RNSCI::
000324  026727  000000G 000014                  CMP     .PSTCN,#4*3     ; LENGTH OK ?
000332  101007                                  BHI     10$             ; IF HI, NO - REJECT TRANSITION
000334  032767  000000G 000000G                 BIT     #WC.CIR,$WILD   ; WILD CARD CIRCUIT ID ?
000342  001003                                  BNE     10$             ; NO !
000344  012702  000000G                         MOV     #$RNSCI,R2      ; DESTINATION
000350  000467                                  BR      STR50           ; STORE RAD50

000352  012767  000000G 000000G 10$:    MOV     #ERRCID,$ERROR
000360  000756                                  BR      REJ
```

```
                                        ;****
                                        ; LOCAL MACROS
                                        ;****

                                                .MACRO  PRINT   TEXT
                                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                                CALL    $PRINT          ; PRINT MESSAGE
                                                .ENDM   PRINT
```

Table of contents

```
     4-   53     MACRO DEFINITIONS
     5-   78     LOCAL DATA
     6-  108     $PUXPR - PURGE MODULE X25-PROTOCOL
     7-  131     PURDTE - PURGE DTE DESCRIPTOR
     8-  205     CHKDTE - CHECK DTE ELIGIBILITY FOR DELETION
     9-  241     PURCUG - PURGE CLOSED USER GROUP
    10-  282     $PUXSR- PURGE MODULE X25-SERVER
    10-  283     $PUX9S- PURGE MODULE X29-SERVER
    11-  378     XSVCTM - PURGE X25-SERVER COUNTER TIMER
    12-  404     X29CTM - PURGE X29-SERVER COUNTER TIMER
    13-  428     $PUXPR - PURGE MODULE X25-ACCESS
    14-  473     TSTPSI - TEST FOR PSI IN SYSTEM
    15-  491     DEALC - REMOVE ENTRY FROM QUEUE
```

```
                                                .SBTTL  $PUXPR - PURGE MODULE X25-ACCESS
                                        ;+
                                        ; *** - $PUXPR - PURGE MODULE X25-PROTOCOL
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = REQUEST CODE
                                        ;
                                        ; OUTPUT:
                                        ;       THE REQUESTED REMOTE DESTINATIONS ARE PURGED.
                                        ;
                                        ;-

001710                                  $PUXAC::
001710  004767  000126                          CALL    TSTPSI          ; MAKE SURE PSI IN SYSTEM
001714  005004                                  CLR     R4              ; NONE FOUND YET
001716  012702  000000G                         MOV     #$TMLST,R2      ; POINT TO LISTHEAD
001722  010205                          10$:    MOV     R2,R5           ; SAVE PREVIOUS ADDRESS
001724  011202                                  MOV     (R2),R2         ; NEXT ADDRESS
001726  001431                                  BEQ     30$             ; BR IF END OF LIST
001730  122762  000000G 000000G                 CMPB    #CS.RDT,C.STS(R2) ; CHECK FOR REMOTE DESTINATION
001736  001371                                  BNE     10$             ; BR UNLESS RDT
001740  032767  000000G 000000G                 BIT     #XA.KND,$OPRND  ; CHECK FOR KNOWN DTES
001746  001010                                  BNE     20$             ; BR IF WILDCARD
001750  026762  000000G 000000G                 CMP     $DST,R$DNAM(R2) ; CHECK FIRST PART OF NAME
001756  001361                                  BNE     10$             ; BR IF WRONG RDT
001760  026762  000002G 000002G                 CMP     $DST+2,R$DNAM+2(R2) ; COMPARE SECOND WORD
001766  001355                                  BNE     10$             ; IF NE, KEEP LOOKING
001770  010604                          20$:    MOV     SP,R4           ; INDICATE WILDCARDS OR FOUND
001772  012700  000000G                         MOV     #R$LENX,R0      ; LENGTH OF RDT BLOCK IN POOL..
001776  012701  000000G                         MOV     #R$DLEN,R1      ; LENGTH OF A RDT TEMPLATE
002002  004767  000056                          CALL    DEALC           ; DEALLOCATE IT
002006  010502                                  MOV     R5,R2           ; RESUME SEARCH CORRECTLY
002010  000744                                  BR      10$             ; CONTINUE
002012  032767  000000G 000000G 30$:    BIT     #XA.KND,$OPRND  ; WILDCARD ?
002020  001002                                  BNE     40$             ; BR IF YES
002022  005704                                  TST     R4              ; ANYTHING FOUND ?
002024  001401                                  BEQ     101$            ; IF EQ, NOPE
002026  000207                          40$:    RETURN
                                        ;
                                        ; ERRORS
                                        ;
002030                                  101$:   ERROR$  #ERRNSD         ; NO SUCH DESTINATION
002040  000207                                  RETURN
```

```
                                          .SBTTL  PURREM - PURGE REMOTE
                                  ;+
                                  ; *** - PURREM - PURGE REMOTE
                                  ;
                                  ; INPUT:
                                  ;       $NODAD = THE REMOTE NODE ADDRESS
                                  ;       $WILD = WILD CARD SPECIFICATIONS
                                  ;
                                  ; OUTPUT:
                                  ;       TEMPLATE IS DELETED
                                  ;
                                  ;-

000006                            $PUREM::
000006  012702  000000G                   MOV     #$TMLST,R2          ; GET START OF TEMPLATES
000012  005067  177762                    CLR     SUCC                ; SUCCESS INDICATOR
                                  ;
                                  ; SEARCH FOR THE REMOTE TEMPLATE
                                  ;
000016  010205                    10$:    MOV     R2,R5               ; COPY CURRENT TEMPLATE ADDRESS
000020  011502                    15$:    MOV     (R5),R2             ; GET NEXT TEMPLATE
000022  001447                            BEQ     30$                 ; IF EQ, END OF LIST
000024  122762  000000G 000000G           CMPB    #CS.REM,C.STS(R2) ; IS THIS A REMOTE TEMPLATE
000032  001371                            BNE     10$                 ; IF NE, NO .. KEEP LOOKING
000034  022767  000000G 000000G           CMP     #WC.NOD,$WILD       ; PURGE ALL REMOTES ?
000042  001405                            BEQ     20$                 ; IF EQ, YES
000044  016204  000004G                   MOV     C.PRM+4(R2),R4      ; SAVE NODE ADDRESS
000050  026704  000000G                   CMP     $NODAD,R4           ; IS THE THE RIGHT TEMPLATE ?
000054  001360                            BNE     10$                 ; IF NE, NO .. KEEP LOOKING
                                  ;
                                  ; DELETE TEMPLATE FROM LIST
                                  ;
000056  005267  177716            20$:    INC     SUCC                ; REMEMBER SUCCESS
000062  032767  000000G 000000G           BIT     #RN.NAM,$OPRND      ; PURGE NAME/ALL ?
000070  001416                            BEQ     25$                 ; IF EQ, LEAVE REMOTE BLOCK INTACT
000072  004767  000612                    CALL    QREM                ; UNLINK TEMPLATE FROM CHAIN
000076  012701  000012                    MOV     #REMLEN,R1          ; GET LENGTH OF A REMOTE TEMPLATE
000102  012700  000000G                   MOV     #$XXAVL,R0          ; GET THE BUFFER LISTHEAD ADDRESS
000106  004767  000000G                   CALL    $RLBF               ; RELEASE CORE BLOCK
000112  162767  000000G 000000G           SUB     #R.LENX,$BYTMN      ; ADJUST MINIMUM POOL BYTE EXTENSION
000120  162767  000000G 000000G           SUB     #R.LENX,$BYTXT      ; AND CURRENT BYTE EXTENSION VALUE
000126  004767  000042            25$:    CALL    PURDLL              ; PURGE DLL COMPONENTS
000132  032767  000000G 000000G           BIT     #WC.NOD,$WILD       ; PURGE ALL REMOTES ?
000140  001327                            BNE     15$                 ; IF NE, YES

000142  005767  177632            30$:    TST     SUCC                ; ANY TEMPLATES FOUND ?
000146  001004                            BNE     40$                 ; IF NE, YES .. OKAY
000150  032767  000000G 000000G           BIT     #WC.NOD,$WILD       ; WAS THIS FOR ALL NODES ?
000156  001401                            BEQ     101$                ; IF EQ, NO .. ERROR
000160  000207                    40$:    RETURN
                                  ;
                                  ; ERROR CONDITIONS
                                  ;
000162                            101$:   ERROR$  #ERRNOD             ; NO REMOTE TEMPLATE FOUND
000172  000207                            RETURN
```

```
                        .TITLE  SETBUF - SET BUFFER PARAMETERS
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - SETUP BUFFER DEFINITION
                ;
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
000242  026727  000000G 000004          CMP     $PRI,#4         ; CHACK AGAINST MINIMUM PRIORITY
000250  103404                          BLO     50$             ; IF LO, ERROR
000252  026727  000000G 000007          CMP     $PRI,#7         ; CHECH AGAINST MAXIMUM PRIORITY
000260  101405                          BLOS    55$             ; IF LOS, YES .. STORE IT
000262                          50$:    ERROR$  #ERRPRI         ; INVALID PRIORITY
000272  000405                          BR      60$             ; AND LOOK FOR MORE WORK
000274  016765  000000G 000012  55$:    MOV     $PRI,C.PRI(R5)  ; ELSE, UPDATE THE PRIORITY
000302  004767  000024                  CALL    UPDDDM          ; UPDATE THE DDM PRIORITY
000306  000676                  60$:    BR      25$             ; LOOK FOR MORE WORK

000310  005701                  65$:    TST     R1              ; SUCCESS ?
000312  001401                          BEQ     70$             ; IF EQ, NO
000314  000207                          RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000316                          70$:    ERROR$  #ERRLIS         ; LINE NOT IN SYSTEM
000326  000261                          SEC                     ; SET CARRY
000330  000207                          RETURN                  ; AND LEAVE
```

```
Symbol table

CS.CHN= ****** GX       C$ULEN= ****** GX       ERRNUM  000100R         XP.NUM= ****** GX       $OPRND= ****** GX
CS.CUG= ****** GX       C$UNAM= ****** GX       EX$ERR= 000002          $BYTMN= ****** GX       $RLBF = ****** GX
CS.DTE= ****** GX       C.STS = ****** GX       G$LENX= ****** GX       $BYTXT= ****** GX       $RQBF = ****** GX
CS.FLG= ****** GX       D$TADR= ****** GX       XP.BIL= ****** GX       $CFERR= ****** GX       $SECUG  000142RG
CS.PVC= ****** GX       ERRALL= ****** GX       XP.DTA= ****** GX       $CUG   = ****** GX      $SWICH= ****** GX
CU$BIL= ****** GX       ERRDTE  000041R         XP.KND= ****** GX       $DTE   = ****** GX      $TMLST= ****** GX
C$UCUG= ****** GX       ERRNSD  000000R         XP.KNG= ****** GX       $NUM   = ****** GX      $XXAVL= ****** GX
C$UFLG= ****** GX

. ABS.  000000     000   (RW,I,GBL,ABS,OVR)
        000634     001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8445 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.14
SY:SETCUG.V2,[132,134]SETCUG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETCUG
```

```
001050  000740                          BR      40$             ; CONTINUE

001052  012605                  60$:    MOV     (SP)+,R5        ; RESTORE R5
001054  000207                          RETURN

```

```
000400  012704  000000G                 MOV     #$KNEVT,R4              ; POINT TO KNOWN EVENTS TABLE
000404  012467  000000G         70$:    MOV     (R4)+,$CLASS            ; GET EVENT CLASS TO PROCESS
000410  100546                          BMI     150$                    ; BR IF END OF TABLE
000412  012467  000000G                 MOV     (R4)+,$EVENT+0          ; GET EVENT MASKS
000416  012467  000002G                 MOV     (R4)+,$EVENT+2          ; ...
000422  012467  000004G                 MOV     (R4)+,$EVENT+4          ; ...
000426  012467  000006G                 MOV     (R4)+,$EVENT+6          ; ...
000432  032767  000000G 000000G 80$:    BIT     #WC.LOG,$WILD           ; KNOWN LOGGING SPECIFIED?
000440  001404                          BEQ     100$                    ; BR IF NO
000442  012703  000000'                 MOV     #SNKTBL,R3              ; POINT TO SINK TYPE TABLE
000446  012305                  90$:    MOV     (R3)+,R5                ; GET SINK TYPE
000450  001522                          BEQ     145$                    ; BR IF END OF TABLE
000452  004767  000316          100$:   CALL    PROEVT                  ; PROCESS THE EVENT
000456  103005                          BCC     105$                    ; BR IF VALID EVENT
000460                          103$:   ERROR$  #ERREVT                 ; DISPLAY ERROR MESSAGE
000470  000516                          BR      150$                    ; AND EXIT
000472  012700  000000G         105$:   MOV     #$TMLST,R0              ; GET START OF TEMPLATES
000476  011000                  110$:   MOV     (R0),R0                 ; GET NEXT TEMPLATE
000500  001430                          BEQ     120$                    ; BR IF NOT IN LIST - CREATE NEW ENTRY
000502  122760  000000G 000000G         CMPB    #CS.EVT,C.STS(R0)       ; IS THIS AN EVENT TEMPLATE?
000510  001372                          BNE     110$                    ; BR IF NO
000512  026760  000000G 000004          CMP     $CLASS,C.CLS(R0)        ; IS THIS THE CORRECT TEMPLATE?
000520  001366                          BNE     110$                    ; BR IF NO
000522  020560  000016                  CMP     R5,C.SNK(R0)            ; IS THIS THE CORRECT TEMPLATE?
000526  001363                          BNE     110$                    ; BR IF NO
                                ;
                                ; FOUND EVENT DEFINITION
                                ;
000530  056760  000000G 000006          BIS     $EVENT+0,C.EVT+0(R0)    ; SET THE EVENT MASKS
000536  056760  000002G 000010          BIS     $EVENT+2,C.EVT+2(R0)    ; ...
000544  056760  000004G 000012          BIS     $EVENT+4,C.EVT+4(R0)    ; ...
000552  056760  000006G 000014          BIS     $EVENT+6,C.EVT+6(R0)    ; ...
000560  000452                          BR      140$
                                ;
                                ; CREATE A NEW TEMPLATE
                                ;
000562  012700  000000G         120$:   MOV     #$XXAVL,R0              ; POINT AT END OF TASK LISTHEAD
000566  012701  000020                  MOV     #C.LEN,R1               ; GET LENGTH OF AN OBJECT TEMPLATE
000572  004767  000000G                 CALL    $RQBF                   ; TRY TO ALLOCATE A TEMPLATE
000576  103005                          BCC     130$                    ; BR IF SUCCESSFUL
000600                                  ERROR$  #ERRBUF                 ; PRINT ALLOCATION FAILURE MESSAGE
000610  000446                          BR      150$                    ; AND EXIT
000612  012760  000000G 000000G 130$:   MOV     #CS.EVT,C.STS(R0)       ; INDICATE EVENT TEMPLATE
000620  016760  000000G 000004          MOV     $CLASS,C.CLS(R0)        ; STORE EVENT CLASS
000626  010560  000016                  MOV     R5,C.SNK(R0)            ; STORE SINK TYPE
000632  016760  000000G 000006          MOV     $EVENT+0,C.EVT+0(R0)    ; STORE EVENT MASKS
000640  016760  000002G 000010          MOV     $EVENT+2,C.EVT+2(R0)    ; ...
000646  016760  000004G 000012          MOV     $EVENT+4,C.EVT+4(R0)    ; ...
000654  016760  000006G 000014          MOV     $EVENT+6,C.EVT+6(R0)    ; ...
000662  004767  000042                  CALL    FINEVT                  ; FIND PLACE FOR TEMPLATE
000666  004767  000000G                 CALL    $QINSN                  ; INSERT ENTRY INTO QUEUE
000672  062767  000000G 000000G         ADD     #F.LENX,$BYTMN          ; UPDATE BYTE AREA MINIMUM
000700  062767  000000G 000000G         ADD     #F.LENX,$BYTXT          ; UPDATE CURRENT BYTE EXTENSION
000706  032767  000000G 000000G 140$:   BIT     #WC.LOG,$WILD           ; KNOWN LOGGING SPECIFIED?
000714  001254                          BNE     90$                     ; BR IF YES
000716  032767  000000G 000000G 145$:   BIT     #OM.KEV,$OPRND          ; KNOWN EVENTS SPECIFIED?
000724  001227                          BNE     70$                     ; BR IF YES
```

```
                                                .SBTTL  $SEXSV- DEFINE MODULE X25-SERVER
                                                .SBTTL  $SEX29- DEFINE MODULE X29-SERVER
                                        ;+
                                        ; *** - $SEXSV- DEFINE MODULE X25-SERVER
                                        ; *** - $SEX29- DEFINE MODULE X29-SERVER
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = REQUEST CODE
                                        ;       $DST   = PARSED DESTINATION NAME
                                        ;
                                        ; OUTPUT:
                                        ;       DESTINATION DESCRIPTORS UPDATED
                                        ;-

                                                .ENABL  LSB

 000634                                 $SEXSV::
 000634 012704  115443                          MOV     #<^RX25>,R4       ; ONLY X25 DESTINATIONS
 000640 012702  001714'                         MOV     #XSVCHA,R2        ; IF COUNTER TIMER OR MAX CIRCUITS
 000644 000404                                  BR      10$               ; JOIN COMMON CODE

 000646                                 $SEX29::
 000646 012704  115447                          MOV     #<^RX29>,R4       ; ONLY X29 DESTINATIONS
 000652 012702  002054'                         MOV     #X29CHA,R2        ; IN CASE COUNTER TIMER OR MAX CIRCUITS
 000656 004767  002106                  10$:    CALL    TSTPSI            ; MAKE SURE THIS IS A PSI SYSTEM
 000662 032767  000000G 000000G                 BIT     #XS.CTM!XS.MXC,$OPRND ; IS THIS CHARACTERISTICS ?
 000670 001401                                  BEQ     20$               ; IF EQ, NO
 000672                                         CALLR   (R2)              ; ELSE DISPATCH TO ROUTINE
                                        ;
                                        ; LOOK FOR DESTINATION DESCRIPTOR
                                        ;
 000674 012702  000000G                 20$:    MOV     #$PSIPT,R2        ; POINT TO LISTHEAD
 000700 005003                                  CLR     R3                ; INITIALLY, NO LINK ADDRESS
 000702 010205                          30$:    MOV     R2,R5             ; SAVE PREVIOUS ADDRESS
 000704 011202                                  MOV     (R2),R2           ; TRAVERSE LIST
 000706 001423                                  BEQ     40$               ; BR IF END OF LIST
 000710 004767  001230                          CALL    CHKDST            ; CHECK FOR DESTINATION TEMPLATE
 000714 103372                                  BCC     30$               ; BR IF NOT DESTINATION BLOCK
 000716 020462  000000G                         CMP     R4,D$STYP(R2)     ; CHECK DESTINATION TYPE
 000722 001367                                  BNE     30$               ; IF NE, WRONG TYPE
 000724 032767  000000G 000000G                 BIT     #XS.KND,$OPRND    ; WILDCARD DESTINATIONS ?
 000732 001020                                  BNE     50$               ; BR IF YES
 000734 026762  000000G 000000G                 CMP     $DST,D$SNAM(R2)   ; COMPARE FIRST WORD OF NAME
 000742 001357                                  BNE     30$               ; BR IF WRONG DESTINATION
 000744 026762  000002G 000002G                 CMP     $DST+2,D$SNAM+2(R2) ; COMPARE SECOND PART OF NAME
 000752 001353                                  BNE     30$               ; IF NE, KEEP LOOKING
 000754 000407                                  BR      50$               ; FOUND IT
                                        ;
                                        ; DESTINATION NOT FOUND
                                        ;
 000756 032767  000000G 000000G         40$:    BIT     #XS.KND,$OPRND    ; WILDCARD OPERATION ?
 000764 001031                                  BNE     60$               ; IF NE, YES - DONE
 000766 004767  001234                          CALL    ALCDST            ; ALLOCATE A DESTINATION TEMPLATE
 000772 103426                                  BCS     60$               ; IF CS, ERROR
                                        ;
                                        ; UPDATE PARAMETERS
                                        ;
```

```
    572                                                     .SBTTL  TSTPSI - TEST FOR PSI IN SYSTEM
    573                                             ;+
    574                                             ;** TSTPSI - TEST FOR PSI IN SYSTEM
    575                                             ;
    576                                             ; INPUTS          $PSIPT = POINTER TO PSN$DF TEMPLATE.
    577                                             ;
    578                                             ; OUTPUT          IF NO PSI SUPPORT, ISSUES ERROR MESSAGE
    579                                             ;                 AND RETURNS TO CALLER'S CALLER.
    580                                             ;-
    581
    582 002770                                  TSTPSI:
    583 002770  005767  000000G                         TST     $PSIPT          ; IS THERE PSI SUPPORT ?
    584 002774  001005                                  BNE     10$             ; IF NE, YES
    585 002776                                          ERROR$  #ERRPSI         ; ERROR - NO PSI SUPPORT
    586 003006  005726                                  TST     (SP)+           ; NO MORE WORK HERE
    587 003010  000207                          10$:    RETURN
    588
    193 000262  056760  000000G 000054  35$:    BIS     $HOST,C.HOS(R0) ; ELSE, STORE NEW HOST
    194 000270  000207                  40$:    RETURN                  ; RETURN TO CALLER
    195                                 ;
    196                                 ; ERROR CONDITIONS
    197                                 ;
    198 000272                          101$:   ERROR$  #ERRNOD         ; NO NODE TEMPLATE
    199 000302  000207                          RETURN
```

$SENSP - SET NSP PARAMETERS

```
000356  032767  000000G 000000G 70$:    BIT     #OE.VFY,$OPRND  ; WAS THE VERIFICATION STATE SPECIFIED ?
000364  001435                          BEQ     90$             ; IF EQ, NO .. LEAVE
                                ;
                                ; SEARCH FOR NSP FEATURES DEFINITION
                                ;
000366  012705  000000G                 MOV     #$TMLST,R5      ; POINT AT START OF TEMPLATES
000372  011505                  80$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
000374  001451                          BEQ     120$            ; IF EQ, END OF LIST
000376  122765  000000G 000000G         CMPB    #CS.FEA,C.STS(R5) ; IS THIS A FEATURES TEMPLATE ?
000404  001372                          BNE     80$             ; IF NE, NO .. KEEP LOOKING
000406  022765  051516  000000G         CMP     #"NS,C.PRM(R5)  ; IS THIS THE NSP FEATURES TEMPLATE ?
000414  001366                          BNE     80$             ; IF NE, NO .. KEEP LOOKING
                                ;
                                ; CHECK IF VERIFICATION EXISTS
                                ;
000416  032765  000001  000002G         BIT     #NF.ACC,C.PRM+2(R5) ; DOES THIS SYSTEM HAVE ACCESS VERIFICATION?
000424  001442                          BEQ     130$            ; IF EQ, NO .. TELL USER
000426  052765  000000G 000000G         BIS     #CS.MOD,C.STS(R5) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000434  042765  100000  000006G         BIC     #NF$ACC,C.PRM+6(R5) ; ASSUME USER DOESN'T WANT ACCESS VERIFICATION
000442  032767  000000G 000000G         BIT     #SW.VFY,$SWICH  ; DOES THE USER WANT ACCESS VERIFICATION ?
000450  001403                          BEQ     90$             ; IF EQ, NO
000452  052765  100000  000006G         BIS     #NF$ACC,C.PRM+6(R5) ; ENFORCE THE ACCESS VERIFICATION
                                ;
                                ; CHECK FOR ECL SEGMENT SIZE
                                ;
000460  032767  000000G 000000G 90$:    BIT     #E2.SEG,$OPRN2  ; SEGMENT BUFFER SIZE SPECIFIED ?
000466  001406                          BEQ     100$            ; IF EQ, NO
000470  052764  000000G 000000G         BIS     #CS.MOD,C.STS(R4) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000476  016764  000000G 000004          MOV     $ESEG,C.SEG(R4) ; ELSE STUFF IT

000504  000416                  100$:   BR      140$            ; RETURN TO CALLER
                                ;
                                ; ERROR CONDITIONS
                                ;
000506                          110$:   ERROR$  #ERRDEC         ; NO DEC TEMPLATE
000516  000411                          BR      140$
000520                          120$:   ERROR$  #ERRFEA         ; NO NSP FEATURES TEMPLATE
000530  000404                          BR      140$
000532                          130$:   ERROR$  #ERRSUP         ; ACCESS VERIFICATION NOT SUPPORTED
000542  012605                  140$:   MOV     (SP)+,R5        ; RESTORE R5
000544  000207                          RETURN

        000001                          .END
```

```
                                                                L  1
PRSMOD     CREATED BY  MACRO  ON 29-JUN-85 AT 05:19     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE              REFERENCES

.MOMC1    000420 RG        #15-281
.MOMDA    000170 RG        #8-150
.MOMRE    000300 RG        #11-208
.MOMXR    000350 RG        #13-244
.MONNA    001252 RG        #27-572
.MONUM    000522 RG        #17-327
.MOOBJ    001176 RG        #26-548
.MOOBN    001152 RG        #25-531
.MOPRI    001050 RG        #23-482
.MORET    000324 RG        #12-226
.MORST    000374 RG        #14-262
.PNUMB  = ******   GX       5-95      5-97      6-113     6-115     7-131     8-153     9-174     9-176     10-193
                            10-195    11-211    11-213    12-229    12-231    13-247    13-249    14-265    14-267
                            15-284    15-286    16-308    17-330    17-332    19-365    23-485    23-487    24-510
                            25-534    25-536
.PNUMH  = ******   GX       5-93      6-111     7-129     8-151     9-172     10-191    11-209    12-227    13-245
                            14-263    15-282    16-302    17-328    19-363    23-483    24-508    25-532
.PSTCN  = ******   GX       4-78      21-420    22-447    22-449    22-452    26-549    26-556    27-573    27-580
.PSTPT  = ******   GX       4-70      21-412    22-457    26-551    27-575
```

M 1

```
                .TITLE  PRSPRO - ACTION ROUTINES FOR PROCESS
                .IDENT  /V05.00/

;
;       COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
;       DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
;       THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
;       ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
;       INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
;       COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
;       OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
;       TRANSFERRED.
;
;       THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
;       AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
;       CORPORATION.
;
;       DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
;       SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
;       MODULE DESCRIPTION:
;
;       CFE - ACTION ROUTINES FOR PROCESS
;
;
;       DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
;       IDENT HISTORY:
;
;       1.00    14-DEC-79
;               DECNET-11M/S V3.0
;               DECNET-11M-PLUS V1.0
;
;       3.00    16-APR-82
;               DECNET-11M V3.1
;               DECNET-11M-PLUS V1.1
;
;       4.00    07-NOV-83
;               DECNET-11M V4.0
;               DECNET-11M-PLUS V2.0
;
;       5.00    22-JUL-85
;               DECnet-11M/S V4.2
;               DECnet-11M-Plus V3.0
;               DECnet-Micro/RSX V1.0
;
;



```

```
                                        .SBTTL  .RNHAR - STORE HARDWARE ADDRESS
                                ;+
                                ; ** .RNHAR - STORE REMOTE NODE HARDWARE ADDRESS
                                ;
                                ; INPUTS:       TPARS CONVENTIONS.
                                ;
                                ; OUTPUT:       $RNHAR  = RAD50 HARDWARE ADDRESS.
                                ;-

        000000                          E       = 0     ; END
        000001                          D       = 1     ; DIGIT
        177777                          H       = -1    ; HYPHEN

                                        .NLIST  BEX

                                ;       DNA FORMAT: XX-XX-XX-XX-XX-XX
000362     021     001     001  T1:     .BYTE   12.+5,  D,D,H,  D,D,H,  D,D,H,  D,D,H,  D,D,H,  D,D,  E

                                ;       RSX FORMAT: XXXXXXXXXXXX
000405     014     001     001  T2:     .BYTE   12.,    D,D,    D,D,    D,D,    D,D,    D,D,    D,D,  E
                                        .EVEN
                                        .LIST   BEX
000424                          .RNHAR::
000424  016700  000000G                 MOV     .PSTPT,R0               ; POINT TO PARSED STRING
000430  012701  000000'                 MOV     #SCR,R1                 ; POINT TO SCRATCH BUFFER
000434  012702  000362'                 MOV     #T1,R2                  ; ASSUME DNA FORMAT
000440  126027  000002  000055          CMPB    2(R0),#<'->             ; HYPHEN PRESENT ?
000446  001402                          BEQ     10$                     ; IF EQ, YES
000450  012702  000405'                 MOV     #T2,R2                  ; ELSE PARSE RSX FORMAT
000454  122267  000000G         10$:    CMPB    (R2)+,.PSTCN            ; CHECK STRING LENGTH
000460  001316                          BNE     REJ                     ; REJECT IF NOT EXACT MATCH
000462  105722                  20$:    TSTB    (R2)+                   ; NEXT CONTROL BYTE
000464  001412                          BEQ     40$                     ; IF EQ, END OF STRING
000466  003004                          BGT     30$                     ; IF GT, HEX DIGIT EXPECTED
000470  122027  000055                  CMPB    (R0)+,#<'->             ; ELSE HYPHEN EXPECTED
000474  001115                          BNE     REJ1                    ; IF NE, ERROR
000476  000771                          BR      20$                     ; ELSE CONTINUE

000500  004767  000100          30$:    CALL    HEX                     ; VALIDATE HEX DIGIT
000504  103511                          BCS     REJ1                    ; REJECT INVALID CHARACTERS
000506  112021                          MOVB    (R0)+,(R1)+             ; COPY CHARACTER
000510  000764                          BR      20$                     ; CONTINUE

000512  012700  000000'         40$:    MOV     #SCR,R0                 ; POINT TO HEX STRING
000516  012701  000014                  MOV     #12.,R1                 ; LENGTH OF HEX STRING
000522  012702  000000G                 MOV     #$RNHAR,R2              ; POINT TO PARSED DATA AREA
000526  000404                          BR      STR50X

                                ;
                                ; STR50 = STORE EXTENDED RAD50 STRING
                                ;
                                ;       R0 = SOURCE
                                ;       R1 = SOURCE LENGTH
                                ;       R2 = DESTINATION
                                ;
000530  016700  000000G         STR50:  MOV     .PSTPT,R0               ; SOURCE
000534  016701  000000G                 MOV     .PSTCN,R1               ; LENGTH
```

```
                                                .SBTTL  .SYPAR - DYNAMIC PARTITION NAME ACTION ROUTINE
                                        ;+
                                        ; *** - .SYPAR - DYNAMIC PARTITION NAME ACTION ROUTINE
                                        ;
                                        ; INPUT:
                                        ;       .PSTPT = ADDRESS OF PARTITION NAME
                                        ;       .PSTCN = COUNT OF CHARS IN PARTITION NAME
                                        ;
                                        ; OUTPUT:
                                        ;       $PARNA = RAD50 PARTITION NAME (2 WORDS)
                                        ;       R0,R1 = DESTROYED
                                        ;       THE TRANSITION IS ACCEPTED OR REJECTED
                                        ;
                                        ;-

000000                                  .SYPAR::
000000  005067  000002G                         CLR     $PARNA+2        ; CLEAR THE 2ND HALF OF THE PARTITION NAME
000004  016700  000000G                         MOV     .PSTPT,R0       ; POINT R0 AT PARTITION NAME
000010  010701                                  MOV     PC,R1           ; PERIODS ARE ACCEPTABLE
000012  004767  000000G                         CALL    $CAT5           ; CONVERT PARTITION NAME TO RAD50
000016  010167  000000G                         MOV     R1,$PARNA       ; STORE THE PARTITION NAME
000022  010701                                  MOV     PC,R1           ; PERIODS ALLOWED
000024  004767  000000G                         CALL    $CAT5           ; CONVERT REST OF PARTITION NAME TO RAD50
000030  010167  000002G                         MOV     R1,$PARNA+2     ; STORE SECOND HALF OF PARTITION NAME
000034  022767  000006  000000G                 CMP     #6,.PSTCN       ; IS THE PARTITION NAME TOO BIG ?
000042  103002                                  BHIS    10$             ; IF HIS, NO - OKAY
000044  062716  000002                          ADD     #2,(SP)         ; ELSE, REJECT THE TRANSITION
000050  000207                          10$:    RETURN
```

```
                .TITLE  PURMOD - PURGE MODULE
                .IDENT  /V05.00/
                .ENABL  LC
;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - PURGE MODULE
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  08-NOV-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
    473                                                 .SBTTL  TSTPSI - TEST FOR PSI IN SYSTEM
    474                                         ;+
    475                                         ; ** TSTPSI - TEST FOR PSI IN SYSTEM
    476                                         ;
    477                                         ; INPUTS        $PSIPT = POINTER TO PSN$DF TEMPLATE.
    478                                         ;
    479                                         ; OUTPUT        IF NO PSI SUPPORT, ISSUES ERROR MESSAGE
    480                                         ;               AND RETURNS TO CALLER'S CALLER.
    481                                         ;-
    482
    483 002042                                  TSTPSI:
    484 002042  005767  000000G                         TST     $PSIPT                  ; IS THERE PSI SUPPORT ?
    485 002046  001005                                  BNE     10$                     ; IF NE, YES
    486 002050                                          ERROR$  #ERRPSI                 ; ERROR - NO PSI SUPPORT
    487 002060  005726                                  TST     (SP)+                   ; NO MORE WORK HERE
    488 002062  000207                          10$:    RETURN
    489
```

```
                                                .SBTTL  PURDLL - PURGE DLL INFO
                                        ;+
                                        ; PURDLL - PURGE REMOTE NODE DLL DATA
                                        ;
                                        ; INPUTS:
                                                R4      = NODE ADDRESS
                                        ;
                                        ; OUTPUT:
                                                DOWNLINE LOAD TEMPLATES AND FILE BLOCKS ARE PURGED.
                                        ;
                                        ;-
 000174                                 PURDLL: SAVRG   R5                      ; SAVE POSITION IN REMOTE LIST SCAN
 000176  012705  000000G                        MOV     #$TMLST,R5              ; POINT TO TEMPLATE LISTHEAD
 000202  000402                                 BR      15$                     ; THIS IS THE FIRST POSSIBLE UNLINK POINT

 000204  011505                         10$:    MOV     (R5),R5                 ; POINT TO NEXT POTENTIAL UNLINK POINT
 000206  001505                                 BEQ     30$                     ; IF EQ, DONE

 000210  011502                         15$:    MOV     (R5),R2                 ; POINT TO NEXT TEMPLATE
 000212  001503                                 BEQ     30$                     ; IF EQ, DONE
 000214  122762  000000G 000000G                CMPB    #CS.SER,C.STS(R2)       ; SER$DF ?
 000222  001370                                 BNE     10$                     ; NO
 000224  020462  000006                         CMP     R4,S$ADD(R2)            ; ADDRESS OK ?
 000230  001365                                 BNE     10$                     ; NO
 000232  010267  177544                         MOV     R2,SERPT                ; SAVE SER$DF POINTER
 000236  011200                                 MOV     (R2),R0                 ; SAVE FIL$DF POINTER
 000240  010067  177540                         MOV     R0,FILPT                ; ...
 000244  122760  000000G 000000G                CMPB    #CS.FIL,C.STS(R0)       ; SANITY CHECK
 000252  001065                                 BNE     40$                     ; TROUBLE

 000254  032767  000000G 000000G                BIT     #RN.NAM,$OPRND          ; DELETING THE WHOLE 9 YARDS ?
 000262  001022                                 BNE     20$                     ; YES
 000264  004767  000142                         CALL    PRGSER                  ; ELSE JUST PLUCK BITS
 000270  004767  000226                         CALL    PRGFIL                  ; AND PURGE SELECTED FILES
                                        ;
                                        ; IF PARTIAL DELETION LEAVES NULL TEMPLATES, DELETE
                                        ; THEM AN UPDATE THE POOL ALLOCATION ACCORDINGLY.
                                        ;
 000274  016702  177502                         MOV     SERPT,R2                ; RECOVER SER$DF ADDRESS
 000300  032762  177776  000004                 BIT     #^CSF.ADD,S$FLG(R2)     ; FLAGS REMAINING ?
 000306  001045                                 BNE     30$                     ; YES - NOT A CANDIDATE FOR DELETION
 000310  016700  177470                         MOV     FILPT,R0                ; POINT TO FILE BLOCK POINTERS
 000314  022020                                 CMP     (R0)+,(R0)+             ; SKIP LINK WORD AND FLAG WORD
 000316  012701  000005                         MOV     #5,R1                   ; 5 FILES
 000322  005720                         17$:    TST     (R0)+                   ; POINTING TO A FILE NAME BLOCK
 000324  001036                                 BNE     30$                     ; IF NE, YES - CAN'T DELETE THIS ONE
 000326  077103                                 SOB     R1,17$                  ; FALL THRU IF BOTH SER$DF, FIL$DF ARE NULL
                                        ;
                                        ; PURGE NODE ALL
                                        ;
 000330  004767  000354                 20$:    CALL    QREM                    ; REMOVE FROM LIST
 000334  012701  000056                         MOV     #S$LEN,R1               ; LENGTH
 000340  012700  000000G                        MOV     #$XXAVL,R0              ; FREELISTHEAD
 000344  004767  000000G                        CALL    $RLBF                   ; RELEASE CORE BLOCK
 000350  162767  000000G 000000G                SUB     #S.LENX,$BYTMN          ; REDUCE MINIMUM POOL BYTE ALLOCATION
 000356  162767  000000G 000000G                SUB     #S.LENX,$BYTXT          ; ..AND CURRENT VALUE
 000364  052767  000000C 000000G                BIS     #RN.SLO!RN.TLO!RN.LOD!RN.DIA!RN.DMP,$OPRND ; PURGE ALL FILES
 000372  004767  000124                         CALL    PRGFIL                  ; GO TO IT
```

```
                                  ;****
                                  ; LOCAL MACROS
                                  ;****

                                          .MACRO  PRINT   TEXT
                                          MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                          CALL    $PRINT          ; PRINT MESSAGE
                                          .ENDM   PRINT

                                          .MACRO  ERROR$  TEXT
                                  ;
                                  ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                  ; MESSAGE STRING.
                                  ;
                                          .IF DIF <TEXT><R0>
                                          MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                          .ENDC
                                          CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                          .ENDM   ERROR$

                                          .MACRO  EPRINT  TEXT
                                  ;
                                  ; PRINT TEXT ON ERROR LUN
                                  ;
                                          .IF DIF <TEXT><R0>
                                          MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                          .ENDC
                                          CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                          .ENDM   EPRINT

                                  ;****
                                  ; MACRO CALLS
                                  ;****
                                          .MCALL  CHADF$

000000                                    CHADF$                  ; DEFINE CHARACTERISTICS WORDS OFFSETS

                                  ;****
                                  ; LOCAL DATA
                                  ;****

000000                            RDBSZ:  .BLKW   1               ; OLD RDB SIZE
000002                            DDM:    .BLKW   1               ; CURRENT DDM PROCESS NAME

                                  ;
                                  ; ERROR MESSAGES
                                  ;
                                          .ENABL  LC
                                          .NLIST  BEX
          000000                          EX$WAR = 0
          000002                          EX$ERR = 2
000004       002     103     106  ERRBUF: .ASCIZ  <EX$ERR>/CFE -- No buffer definition/
000041       000     103     106  ERRTHS: .ASCIZ  <EX$WAR>/CFE -- Minimum receive buffer value changed to %D/
                                          .EVEN
                                          .LIST   BEX

                                  ;
```

```
                                                .SBTTL  UPDDDM - UPDATE THE DDM PRIORITY
                                        ;+
                                        ; ** UPDDDM - UPDATE THE DDM PRIORITY
                                        ;
                                        ; INPUTS        R0 = ADDRESS OF DDM TEMPLATE
                                        ;               R5 = ADDRESS OF CONTROLLER TEMPLATE
                                        ;
                                        ; OUTPUT        THE DDM PRIORITY IS SET TO THAT OF THE CONTROLLER
                                        ;               WITH THE HIGHEST PRIORITY.
                                        ;
                                        ;-

000332                                  UPDDDM:
000332  004267  000000G                         JSR     R2,$SAVVR       ; SAVE VOLATILE REGISTERS
000336  010001                                  MOV     R0,R1           ; COPY DDM POINTER
000340  005002                                  CLR     R2              ; NO PRIORITY, YET

000342  011101                          10$:    MOV     (R1),R1         ; NEXT TEMPLATE
000344  001412                                  BEQ     20$             ; BR IF END OF LIST
000346  122761  000000G 000000G                 CMPB    #CS.CNT,C.STS(R1) ; IS THIS A CONTROLLER TEMPLATE ?
000354  001006                                  BNE     20$             ; IF NE, NO - DO UPDATE
000356  026102  000012                          CMP     C.PRI(R1),R2    ; COMPARE TO MAXIMUM PRIORITY
000362  003767                                  BLE     10$             ; IF LE, CONTINUE
000364  016102  000012                          MOV     C.PRI(R1),R2    ; ELSE UPDATE MAXIMUM
000370  000764                                  BR      10$             ; CONTINUE

000372  010260  000004G                 20$:    MOV     R2,C.PRM+4(R0)  ; ALWAYS SET DDM PRIORITY
000376  052760  000000G 000000G                 BIS     #CS.MOD,C.STS(R0) ; AND MARK AS CHANGED
000404  000207                                  RETURN


        000001                                  .END
```

```
                                                                        L 10
SETCUG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21        PAGE 1
SYMBOL CROSS REFERENCE                                     CREF    04.00
SYMBOL  VALUE              REFERENCES

CS.CHN  = ******   GX      6-185
CS.CUG  = ******   GX      6-119      6-143      6-189
CS.DTE  = ******   GX      6-164
CS.FLG  = ******   GX      6-155
CS.PVC  = ******   GX      6-187
CU$BIL  = ******   GX      6-154
C$UCUG  = ******   GX     *6-151
C$UFLG  = ******   GX     *6-146     *6-154
C$ULEN  = ******   GX      6-138      6-203
C$UNAM  = ******   GX      6-123      6-125     *6-144     *6-145
C.STS   = ******   GX      6-119     *6-143     *6-155      6-164      6-185      6-187      6-189
D$TADR  = ******   GX      6-167
ERRALL  = ******   GX      6-211
ERRDTE    000041 R        #5-93       6-207
ERRNSD    000000 R        #5-92       6-199
ERRNUM    000100 R        #5-94       6-209
EX$ERR  = 000002          #5-91       5-92       5-93       5-94
G$LENX  = ******   GX      6-141      6-142
XP.BIL  = ******   GX      6-152
XP.DTA  = ******   GX      6-133      6-156
XP.KND  = ******   GX      6-121
XP.KNG  = ******   GX      6-131
XP.NUM  = ******   GX      6-135      6-149
$BYTMN  = ******   GX     *6-141
$BYTXT  = ******   GX     *6-142
$CFERR  = ******   GX      6-199      6-207      6-209      6-211
$CUG    = ******   GX      6-123      6-125      6-144      6-145
$DTE    = ******   GX      6-168
$NUM    = ******   GX      6-151
$OPRND  = ******   GX      6-121      6-131
$RLBF   = ******   GX      6-205
$RQBF   = ******   GX      6-139
$SECUG    000142 RG       #6-111
$SWICH  = ******   GX      6-133      6-135      6-149      6-152      6-156
$TMLST  = ******   GX      6-115      6-161     *6-181     *6-193
$XXAVL  = ******   GX      6-137      6-202
```

M 10

```
    327                                                 .SBTTL  DSTATE - CHANGE DTE STATE
    328                                         ;+
    329                                         ; **-DSTATE - CHANGE DTE STATE
    330                                         ;
    331                                         ; INPUTS:       R5 = DTE TEMPLATE ADDRESS
    332                                         ;               $SWICH = PARSED OPTIONS
    333                                         ;
    334                                         ; OUTPUT:       THE DTE STATE IS CHANGED. IF
    335                                         ;               THE REQUESTED STATE IS ON, AND THERE
    336                                         ;               IS ALREADY ANOTHER ACTIVE DTE ON THE SAME LINE
    337                                         ;               THAN AN ERROR MESSAGE IS PRINTED.
    338                                         ;
    339                                         ; REGISTERS:    R0, R1 AND R2 ARE CLOBBERED.
    340                                         ;-
    341
    342 001056                                  DSTATE:
    343 001056  032767  000000G 000000G                 BIT     #XP.ST1,$SWICH  ; REQUESTED STATE = ON ?
    344 001064  001452                                  BEQ     20$             ; NO
    345 001066  016700  000000G                         MOV     $TMLST,R0       ; POINT TO TEMPLATE LISTHEAD
    346 001072  011000                          10$:    MOV     (R0),R0         ; NEXT TEMPLATE
    347 001074  001446                                  BEQ     20$             ; IF EQ END OF LIST
    348 001076  126027  000000G 000000G                 CMPB    C.STS(R0),#CS.DTE ; IS THIS A DTE TEMPLATE ?
    349 001104  001372                                  BNE     10$             ; IF NE, NO
    350 001106  020005                                  CMP     R0,R5           ; SAME AS TARGET DTE ?
    351 001110  001770                                  BEQ     10$             ; YES- DON'T CARE
    352 001112  005760  000000G                         TST     D$TSTA(R0)      ; STATE = ON ?
    353 001116  001765                                  BEQ     10$             ; NO
    354 001120  010001                                  MOV     R0,R1           ; COPY CURRENT DTE TEMPLATE POINTER
    355 001122  062701  000000G                         ADD     #D$TLIN,R1      ; POINT TO LINE ID
    356 001126  010502                                  MOV     R5,R2           ; COPY TARGET DTE TEMPLATE POINTER
    357 001130  062702  000000G                         ADD     #D$TLIN,R2      ; POINT TO LINE ID
    358         000004                                  .REPT   4
    359                                                 CMP     (R1)+,(R2)+     ; COMPARE LINE IDS
    360                                                 BNE     10$             ; IF NE, DIFFERENT LINES
    361                                                 .ENDR
    362 001154  010502                                  MOV     R5,R2           ; ELSE POINT TO TEMPLATE
    363 001156  062702  000000G                         ADD     #D$TADR,R2      ; POINT TO DTE ADDRESS
    364 001162  012701  000052'                         MOV     #ERRLIN,R1      ; POINT TO ERROR STRING
    365 001166  012700  000000G                         MOV     #$BUFF,R0       ; POINT TO OUTPUT BUFFER
    366 001172  112120                                  MOVB    (R1)+,(R0)+     ; OUTPUT SEVERITY CODE
    367 001174  004767  000000G                         CALL    $EDMSG          ; FORMAT THE DTE ADDRESS
    368 001200                                          ERROR$  #$BUFF          ; ERROR
    369 001210  000410                                  BR      30$             ; DONE
    370
    371 001212  005065  000000G                 20$:    CLR     D$TSTA(R5)      ; ASSUME OFF
    372 001216  032767  000000G 000000G                 BIT     #XP.ST1,$SWICH  ; TEST FOR ON STATE
    373 001224  001402                                  BEQ     30$             ; BR IF OFF
    374 001226  005265  000000G                         INC     D$TSTA(R5)      ; ELSE, STATE = ON
    375
    376 001232  000207                          30$:    RETURN
    377
    378         000001                                  .END
    258 000726  000207                  150$:    RETURN
```

```
000774  004767  000066            50$:    CALL    SETDST              ; SET DST$DF PARAMETERS
001000  011202                            MOV     (R2),R2             ; POINT TO DSA$DF TEMPLATE
001002  001423                            BEQ     70$                 ; IF EQ, BIG TROUBLE
001004  122762  000000G 000000G           CMPB    #CS.DSA,C.STS(R2);  CHECK TEMPLATE TYPE
001012  001017                            BNE     70$                 ; IF NE, INTERNAL PROBLEM
001014  004767  000150                    CALL    SETDSA              ; SET DSA$DF PARAMETERS
001020  011202                            MOV     (R2),R2             ; POINT TO DSC$DF
001022  001413                            BEQ     70$                 ; IF EQ, ERROR
001024  122762  000000G 000000G           CMPB    #CS.DSC,C.STS(R2);  CHECK TYPE
001032  001007                            BNE     70$                 ; IF NE, ERROR
001034  004767  000342                    CALL    SETDSC              ; SET DSC$DF PARAMETERS
001040  032767  000000G 000000G           BIT     #XS.KND,$OPRND      ; KNOWN DESTINATIONS ?
001046  001315                            BNE     30$                 ; CONTINUE IF WILDCARDS

001050  000207                    60$:    RETURN
                                  ;
                                  ; ERRORS
                                  ;
001052                            70$:    ERROR$  #ERRCON             ; INVALID DESTINATION SEQUENCE
001062  000167  000000G                   JMP     .CONER              ; THIS IS SERIOUS

                                          .DSABL  LSB

```

```
                                        .SBTTL  RQCB - REQUEST CORE BLOCK
                                ;+
                                ; RQCB - REQUEST CORE BLOCK
                                ;
                                ; INPUTS:       R0 = ADDRESS OF FREE LISTHEAD
                                ;               R1 = LENGTH REQUESTED
                                ;
                                ; OUTPUT: CARRY SET: ALLOCATION FAILURE
                                ;         CARRY CLEAR:
                                ;               R0 = ADDRESS OF ZEROED BLOCK
                                ;               R1 = ACTUAL LENGTH ALLOCATED
                                ;
                                ;
                                ;-

 003012                         RQCB:
 003012  004767  000000G                CALL    $RQBF           ; REQUEST CORE BLOCK
 003016  103407                         BCS     20$             ; IF CS, ALLOCATION FAILURE
 003020  010046                         MOV     R0,-(SP)        ; PUSH REGISTERS
 003022  010146                         MOV     R1,-(SP)        ; ...
 003024  105020                 15$:    CLRB    (R0)+           ; ZERO OUT THE BLOCK
 003026  005301                         DEC     R1              ; ...
 003030  003375                         BGT     15$             ; ...
 003032  012601                         MOV     (SP)+,R1        ; POP REGISTERS
 003034  012600                         MOV     (SP)+,R0        ; ...
 003036  000207                 20$:    RETURN


         000001                         .END
```

```
                                                .SBTTL  CHKADD - CHECK EXECUTOR NODE
                                        ;+
                                        ; *** - CHKADD - CHECK EXECUTOR NODE ADDRESS
                                        ;
                                        ; INPUTS:
                                        ;         $NODAD = EXECUTOR NODE ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;         C-BIT = SUCCESS/FAILURE
                                        ;
                                        ;-

000304                                  CHKADD:
000304  004767  000000G                         CALL    $SAVAL                  ; SAVE ALL REGISTERS
000310  012700  000000G                         MOV     #$TMLST,R0              ; GET START OF TEMPLATES
                                        ;
                                        ; SEARCH FOR THE ROUTING TEMPLATE
                                        ;
000314  011000                          10$:    MOV     (R0),R0                 ; GET NEXT TEMPLATE
000316  001414                                  BEQ     20$                     ; IF EQ, END OF LIST .. THAT'S OKAY
000320  122760  000000G 000000G                 CMPB    #CS.ROU,C.STS(R0)       ; ELSE, IS THIS A ROUTING TEMPLATE ?
000326  001372                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
000330  005760  000004                          TST     C.ADD(R0)               ; IS THIS A ROUTING NODE?
000334  001405                                  BEQ     20$                     ; BR IF NO
000336  026760  000000G 000004                  CMP     $NODAD,C.ADD(R0)        ; IS THE NODE ADDRESS IN RANGE ?
000344  000261                                  SEC                             ; ASSUME NOT
000346  003001                                  BGT     30$                     ; IF GT, NO
000350  000241                          20$:    CLC                             ; ELSE, INDICATE SUCCESS
000352  000207                          30$:    RETURN
```

```
                                                                     L 16
SETNSP - SET NSP PARAMETERS     MACRO V05.03b Saturday 29-Jun-85 05:22  Page 5-2
Symbol table

CD.LEN= 000022          C.IPL   000020          C.SEG   000004          NF.ACC= 000001          $CFERR= ****** GX
CL.LEN= 000020          C.LIN   000011          C.STS = ****** GX       NF.BCP= 000040          $EINA = ****** GX
CS.DEC= ****** GX       C.LKS   000010          ERRDEC  000000R         NF.END= 000200          $EINC = ****** GX
CS.FEA= ****** GX       C.LST   000021          ERRFEA  000030R         NF.EVT= 000002          $EOUT = ****** GX
CS.MOD= ****** GX       C.NAM   000004          ERRSUP  000066R         NF.LLI= 000004          $ERET = ****** GX
C.CTM   000016          C.NCTD= 000013          EX$ERR= 000002          NF.LV2= 000100          $ESEG = ****** GX
C.DELF  000016          C.NDC   000006          E2.INA= ****** GX       NF.SLI= 000400          $LINKS= ****** GX
C.DELW  000017          C.NLT   000014          E2.INC= ****** GX       NF.SMC= 000020          $NCOU = ****** GX
C.DEV   000012          C.NLTD= 000012          E2.OUT= ****** GX       NF.SWP= 000010          $OPRND= ****** GX
C.EXT   000015          C.OUTT  000013          E2.RET= ****** GX       OE.LNK= ****** GX       $OPRN2= ****** GX
C.FLG   000006          C.PRI   000010          E2.SEG= ****** GX       OE.NCO= ****** GX       $SENSP  000140RG
C.INAC  000014          C.PRM = ****** GX       NF$ACC= 100000          OE.VFY= ****** GX       $SWICH= ****** GX
C.INCT  000012          C.RETF  000015          NF$EVT= 040000          SW.VFY= ****** GX       $TMLST= ****** GX


. ABS.  000022    000   (RW,I,GBL,ABS,OVR)
        000546    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9236 Words  ( 37 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.02
SY:SETNSP.V2,[132,134]SETNSP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETNSP
```

```
                                                             M 16
SETNSP     CREATED BY  MACRO  ON 29-JUN-85 AT 05:22     PAGE 1
```

```
**FILE**ID**PRSOBJ




PPPPPPPP     RRRRRRRR       SSSSSSSS     000000     BBBBBBBB             JJ
PPPPPPPP     RRRRRRRR       SSSSSSSS     000000     BBBBBBBB             JJ
PP      PP   RR      RR   SS           00      00   BB      BB           JJ
PP      PP   RR      RR   SS           00      00   BB      BB           JJ
PP      PP   RR      RR   SS           00      00   BB      BB           JJ
PP      PP   RR      RR   SS           00      00   BB      BB           JJ
PPPPPPPP     RRRRRRRR       SSSSSS     00      00   BBBBBBBB             JJ
PPPPPPPP     RRRRRRRR       SSSSSS     00      00   BBBBBBBB             JJ
PP           RR  RR               SS   00      00   BB      BB   JJ      JJ
PP           RR  RR               SS   00      00   BB      BB   JJ      JJ
PP           RR    RR             SS   00      00   BB      BB   JJ      JJ        ....
PP           RR    RR             SS   00      00   BB      BB   JJ      JJ        ....
PP           RR      RR   SSSSSSSS       000000     BBBBBBBB       JJJJJJ          ....
PP           RR      RR   SSSSSSSS       000000     BBBBBBBB       JJJJJJ          ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

N 1

```
                                             .SBTTL  .PRNAM - PROCESS NAME ACTION ROUTINE
                                     ;+
                                     ; *** - .PRNAM - PROCESS NAME ACTION ROUTINE
                                     ;
                                     ; INPUT:
                                     ;       .PSTPT = ADDRESS OF PROCESS NAME
                                     ;       .PSTCN = COUNT OF CHARS IN PROCESS NAME
                                     ;
                                     ; OUTPUT:
                                     ;       $PRNAM = RAD50 PROCESS NAME
                                     ;       R0,R1 = DESTROYED
                                     ;       THE TRANSITION IS ACCEPTED OR REJECTED
                                     ;
                                     ;-

000000                               .PRNAM::
000000  016700  000000G                      MOV     .PSTPT,R0       ; POINT R0 AT PROCESS NAME
000004  010701                               MOV     PC,R1           ; PERIODS ARE ACCEPTABLE
000006  004767  000000G                      CALL    $CAT5           ; CONVERT PROCESS NAME TO RAD50
000012  010167  000000G                      MOV     R1,$PRNAM       ; STORE THE PROCESS NAME
000016  022767  000003  000000G              CMP     #3,.PSTCN       ; IS THE PROCESS NAME TOO BIG ?
000024  103411                               BLO     10$             ; IF LO, YES .. ERROR
000026  012700  000000G                      MOV     #$TMLST,R0      ; POINT TO START OF TEMPLATES
000032  004767  000000G                      CALL    $FNPRO          ; TRY TO FIND AT LEAST 1 PROCESS
000036  103011                               BCC     30$             ; IF CC, FOUND ONE
000040  012767  000000G 000000G              MOV     #ERRPRC,$ERROR  ; ELSE, PROCESS NOT IN SYSTEM
000046  000403                               BR      20$             ; AND REJECT
000050  012767  000000G 000000G 10$:         MOV     #ERRPRO,$ERROR  ; INVALID PROCESS ID
000056  062716  000002          20$:         ADD     #2,(SP)         ; ELSE, REJECT THE TRANSITION
000062  000207                  30$:         RETURN
```

```
000540                          STR50X: SAVRG   R3              ; SAVE REGISTER
000542  010203                          MOV     R2,R3           ; POINT TO PARSED DATA AREA
000544  060001                          ADD     R0,R1           ; POINT PAST STRING
000546  010146                          MOV     R1,-(SP)        ; SAVE ADDRESS
000550  111146                          MOVB    (R1),-(SP)      ; SAVE CHARACTER
000552  112711  000077                  MOVB    #<'?>,(R1)      ; INSERT STOP CHARACTER
000556  004767  000000G         10$:    CALL    .CAT5E          ; CONVERT TO RAD50 (EXTENDED WITH '-')
000562  103402                          BCS     20$             ; IF CS, NO DATA
000564  010123                          MOV     R1,(R3)+        ; STORE CONVERTED VALUE
000566  000773                          BR      10$             ; IF CC, MORE
000570  005701                  20$:    TST     R1              ; ANY FRACTIONAL WORDS ?
000572  001401                          BEQ     30$             ; IF EQ, NOPE
000574  010123                          MOV     R1,(R3)+        ; ELSE STORE LAST ONE
000576  112636                  30$:    MOVB    (SP)+,@(SP)+    ; RESTORE TRAILING CHARACTER
000600                                  RESRG   R3              ; RESTORE REGISTER
000602  000207                          RETURN

                                ;
                                ; HEX - VALIDATE HEX DIGIT
                                ;
                                ; INPUT:        R0 = ADDRESS OF HEX DIGIT
                                ;
                                ; OUTPUT:       CARRY BIT: SUCCESS/FAILURE.
                                ;
000604  121027  000060          HEX:    CMPB    (R0),#'0        ; DIGIT
000610  103412                          BLO     20$             ; IF LO, NO
000612  121027  000071                  CMPB    (R0),#'9        ; REALLY DIGIT ?
000616  101406                          BLOS    10$             ; YES - OK
000620  121027  000101                  CMPB    (R0),#'A        ; HEX DIGIT ?
000624  103404                          BLO     20$             ; IF LO, NO
000626  121027  000106                  CMPB    (R0),#'F        ; REALLY HEX DIGIT ?
000632  101001                          BHI     20$             ; IF HI, ERROR
000634  005727                  10$:    TST     (PC)+           ; SUCCESS
000636  000261                  20$:    SEC                     ; FAILURE
000640  000207                          RETURN
```

```
                                        .SBTTL  .SYPOL - POOL NAME ACTION ROUTINE
                                ;+
                                ; *** - .SYPOL - POOL NAME ACTION ROUTINE
                                ;
                                ; INPUT:
                                ;       .PSTPT = ADDRESS OF POOL NAME
                                ;       .PSTCN = COUNT OF CHARS IN POOL NAME
                                ;
                                ; OUTPUT:
                                ;       $POLNA = RAD50 POOL NAME (2 WORDS)
                                ;       R0,R1 = DESTROYED
                                ;       THE TRANSITION IS ACCEPTED OR REJECTED
                                ;
                                ;-

000052                          .SYPOL::
000052  005067  000002G                 CLR     $POLNA+2        ; CLEAR THE SECOND HALF OF THE POOL NAME
000056  016700  000000G                 MOV     .PSTPT,R0       ; POINT R0 AT POOL NAME
000062  010701                          MOV     PC,R1           ; PERIODS ARE ACCEPTABLE
000064  004767  000000G                 CALL    $CAT5           ; CONVERT POOL NAME TO RAD50
000070  010167  000000G                 MOV     R1,$POLNA       ; STORE THE POOL NAME
000074  010701                          MOV     PC,R1           ; PERIODS ALLOWED
000076  004767  000000G                 CALL    $CAT5           ; CONVERT REST OF POOL NAME TO RAD50
000102  010167  000002G                 MOV     R1,$POLNA+2     ; STORE SECOND HALF OF POOL NAME
000106  022767  000006  000000G         CMP     #6,.PSTCN       ; IS THE POOL NAME TOO BIG ?
000114  103002                          BHIS    10$             ; IF HIS, NO - OKAY
000116  062716  000002                  ADD     #2,(SP)         ; ELSE, REJECT THE TRANSITION
000122  000207                  10$:    RETURN
```

```
                        .SBTTL  MACRO DEFINITIONS

                ;****
                ; LOCAL MACROS
                ;****

                        .MACRO  PRINT   TEXT
                        MOV     TEXT,R0                 ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                        CALL    $PRINT                  ; PRINT MESSAGE
                        .ENDM   PRINT

                        .MACRO  ERROR$  TEXT
                ;
                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                ; MESSAGE STRING.
                ;
                        .IF DIF <TEXT><R0>
                        MOV     TEXT,R0                         ; GET ADDRESS OF ERROR MESSAGE
                        .ENDC
                        CALL    $CFERR                          ; PRINT ERROR MESSAGE
                        .ENDM   ERROR$

                        .MCALL  CALLR
```

```
                                        .SBTTL  DEALC - REMOVE ENTRY FROM QUEUE
                                ;+
                                ; *** - DEALC - REMOVE ENTRY FROM QUEUE
                                ;
                                ; INPUTS:
                                ;       R5 = PREVIOUS ENTRY IN QUEUE
                                ;       R2 = ENTRY TO REMOVE
                                ;       R1 = SIZE OF ENTRY
                                ;       R0 = SIZE OF ENTRY IN POOL BYTE EXTENSION
                                ;       $TMLST = QUEUE LISTHEAD
                                ;       $XXAVL = FREE LISTHEAD
                                ;
                                ; OUTPUTS:
                                ;       R5,R4,R3 ARE PRESERVED
                                ;       R2 = NEXT NODE IN LIST
                                ;       ENTRY IS REMOVED FROM QUEUE AND RETURNED TO THE FREE LIST.
                                ;
                                ;-

002064                          DEALC:
002064  032700  000003          1$:     BIT     #3,R0           ; HAVE TO ROUND UP TO 4 BYTE BOUNDARY
002070  001402                          BEQ     5$              ; IF EQ, ALREADY THERE
002072  005200                          INC     R0              ; ADD A BYTE
002074  000773                          BR      1$              ; TRY AGAIN
002076  160067  000000G         5$:     SUB     R0,$BYTMN       ; UPDATE POOL BYTE MINIMUM
002102  160067  000000G                 SUB     R0,$BYTXT       ; UPDATE ACTUAL ALLOCATION
002106  011215                          MOV     (R2),(R5)       ; CLOSE UP LIST
002110  001002                          BNE     10$             ; IF NE, NOT LAST IN LIST
002112  010567  000002G                 MOV     R5,$TMLST+2     ; SET NEW LAST ENTRY
002116  012700  000000G         10$:    MOV     #$XXAVL,R0      ; POINT TO FREE LIST
002122  004767  000000G                 CALL    $RLBF           ; RELEASE CORE BLOCK
002126  011502                          MOV     (R5),R2         ; POINT TO NEW NEXT NODE
002130  000207                          RETURN

        000001                          .END
```

```
    000376  016702  177402                  MOV     FILPT,R2        ; RECOVER TEMPLATE ADDRESS
    000402  004767  000302                  CALL    QREM            ; REMOVE FROM LIST
    000406  012701  000016                  MOV     #F$LEN,R1       ; FIL$DF LENGTH
    000412  012700  000000G                 MOV     #$XXAVL,R0      ; FREELISTHEAD
    000416  004767  000000G                 CALL    $RLBF           ; RELEASE CORE BLOCK

    000422                          30$:    RESRG   R5              ; RESTORE POINTER TO REMOTE BLOCKS
    000424  000207                          RETURN

    000426  000167  000000G         40$:    JMP     .CONER
```

```
                                ; DEFINE THE ASYNCHRONOUS BAUD RATE TABLE
                                ;
000124                          RATETB:                         ; BIT SETTINGS
000124  000062                          .WORD   50.             ; 0
000126  000113                          .WORD   75.             ; 1
000130  000156                          .WORD   110.            ; 2
000132  000206                          .WORD   134.            ; 3
000134  000226                          .WORD   150.            ; 4
000136  000454                          .WORD   300.            ; 5
000140  001130                          .WORD   600.            ; 6
000142  002260                          .WORD   1200.           ; 7
000144  003410                          .WORD   1800.           ; 10
000146  003720                          .WORD   2000.           ; 11
000150  004540                          .WORD   2400.           ; 12
000152  007020                          .WORD   3600.           ; 13
000154  011300                          .WORD   4800.           ; 14
000156  016040                          .WORD   7200.           ; 15
000160  022600                          .WORD   9600.           ; 16
                                                                ; 17 - EXTERNAL CLOCK (SYNC)
                                ;
                                ; DEFINE THE BUFFER BUFFER BLOCK
                                ;
000000                                  .ASECT
        000000                          .=0
000000                                  .BLKW   1                       ; C.LNK
000002                                  .BLKW   1                       ; C.STS
000004                          C.CBNM: .BLKW   1                       ; CCB NUMBER
000006                          C.CBSZ: .BLKW   1                       ; CCB SIZE
000010                          C.RBNM: .BLKW   1                       ; RDB NUMBER
000012                          C.RBSZ: .BLKW   1                       ; RDB SIZE
000014                          C.SBNM: .BLKW   1                       ; SDB NUMBER
000016                          C.SBSZ: .BLKW   1                       ; SDB NUMBER
000020                          C.THSH: .BLKW   1                       ; RDB THRESHOLD VALUE
                                ;
                                ; DEFINE THE UNIT TEMPLATE
                                ;
        000000                          .=0
000000                                  .BLKW   1                       ; C.LNK
000002                                  .BLKW   1                       ; C.STS
000004                          C.UNT:  .BLKW   1                       ; UNIT NUMBER
000006                          C.CW0:  .BLKW   1                       ; CHARACTERISTICS WORD 0
000010                          C.CW1:  .BLKW   1                       ; CHARACTERISTICS WORD 1
000012                          C.CSR:  .BLKW   1                       ; SECONDARY CSR ADDRESS
000162                                  .PSECT
```

```
                                                                                      M  9
SETCNT - SET CONTROLLER OPTIONS MACRO V05.03b Saturday 29-Jun-85 05:21  Page 6-1
Symbol table

CS.CNT= ****** GX        C.PRM = ****** GX        OL.CSR= ****** GX        $CFERR= ****** GX        $SAVVR= ****** GX
CS.DDM= ****** GX        C.STS = ****** GX        OL.PRI= ****** GX        $CNTNM= ****** GX        $SECNT  000040RG
CS.MOD= ****** GX        C.URM   000014           OL.URM= ****** GX        $CSR   = ****** GX       $TMLST= ****** GX
C.CNT   000004           C.VEC   000006           OL.VEC= ****** GX        $DVNAM= ****** GX        $URM   = ****** GX
C.CSR   000010           ERRLIS= ****** GX        UPDDDM  000332R          $OPRND= ****** GX        $VEC   = ****** GX
C.LEN = 000016           ERRPRI  000000R          WC.CNT= ****** GX        $PRI   = ****** GX       $WILD  = ****** GX
C.PRI   000012           EX$ERR= 000002           WC.NAM= ****** GX

. ABS.  000016    000    (RW,I,GBL,ABS,OVR)
        000406    001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8435 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.49
SY:SETCNT.V2,[132,134]SETCNT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETCNT
```

N 9

```
SETCUG     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

EPRINT          #4-74
ERROR$          #4-63       6-199      6-207      6-209      6-211
PRINT           #4-58
```

```
SETDTE - SET LOCAL DTE PARAMETE MACRO V05.03b Saturday 29-Jun-85 05:21  Page 10-1
Symbol table

CH$HI = ****** GX      C$HLEN= ****** GX      ERRPMI  000012R        XP.DTE= ****** GX      $DVNAM= ****** GX
CS.CHN= ****** GX      C$HLO = ****** GX      EX$ERR= 000002         XP.KND= ****** GX      $EDMSG= ****** GX
CS.CUG= ****** GX      C.STS = ****** GX      EX$WAR= 000000         XP.LIN= ****** GX      $FLAGS= ****** GX
CS.DTE= ****** GX      DEFHSH= 000040         FL.MUX= ****** GX      XP.MCI= ****** GX      $MCI  = ****** GX
CS.PSN= ****** GX      DSTATE  001056R        FL.TRB= ****** GX      XP.STA= ****** GX      $OPRND= ****** GX
CS.PVC= ****** GX      D$TADR= ****** GX      FL.TRI= ****** GX      XP.ST1= ****** GX      $PSIPT= ****** GX
CS.SVC= ****** GX      D$TCT = ****** GX      INSDTE  000542R        $BUFF = ****** GX      $RLBF = ****** GX
CS.TRB= ****** GX      D$THSH= ****** GX      L$LENX= ****** GX      $BYTMN= ****** GX      $RQBF = ****** GX
CS.UNI= ****** GX      D$TLEN= ****** GX      PRDLST  000000R        $BYTXT= ****** GX      $SEDTE  000134RG
CS.X2P= ****** GX      D$TLIN= ****** GX      SETCHN  000674R        $CFERR= ****** GX      $SWICH= ****** GX
CS.X29= ****** GX      D$TSTA= ****** GX      SETLIN  000606R        $CHNLS= ****** GX      $TMLST= ****** GX
CS.X3P= ****** GX      ERRALL= ****** GX      XP.CHN= ****** GX      $CT   = ****** GX      $XXAVL= ****** GX
C$HHI = ****** GX      ERRLIN  000052R        XP.COU= ****** GX      $DTE  = ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001234    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8480 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.15
SY:SETDTE.V2,[132,134]SETDTE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETDTE
```

```
                                                .SBTTL  FINEVT - FIND NEXT EVENT IN LIST

                                        ;+
                                        ;
                                        ; FINEVT - FIND NEXT EVENT IN SORTED LIST
                                        ;
                                        ; INPUTS:
                                        ;       R0 = TEMPLATE TO BE INSERTED
                                        ;
                                        ; OUTPUTS:
                                        ;       R1 = ADRESS OF TEMPLATE TO PRECEDE CURRENT TEMPLATE
                                        ;
                                        ;-
 000730  012702  000000G                FINEVT: MOV     #$TMLST,R2              ; GET START OF TEMPLATES
 000734  010201                         10$:    MOV     R2,R1                   ; SAVE ADDRESS OF PREVIOUS
 000736  011202                                 MOV     (R2),R2                 ; GET NEXT TEMPLATE IN LIST
 000740  001414                                 BEQ     20$                     ; BR IF END OF LIST
 000742  122762  000000G 000000G                CMPB    #CS.EVT,C.STS(R2)       ; IS THIS AN EVENT TEMPLATE?
 000750  001371                                 BNE     10$                     ; BR IF NO
 000752  026062  000004  000004                 CMP     C.CLS(R0),C.CLS(R2)     ; DOES TEMPLATE GO HERE?
 000760  101365                                 BHI     10$                     ; BR IF NO
 000762  026062  000016  000016                 CMP     C.SNK(R0),C.SNK(R2)     ; DOES TEMPLATE GO HERE?
 000770  101361                                 BHI     10$                     ; BR IF NO
 000772  000207                         20$:    RETURN
```

```
                                                .SBTTL  SETDST - SET DST$DF PARAMETERS
                                        ;+
                                        ; ** SETDST - SET DST$DF PARAMETERS
                                        ;
                                        ; INPUTS        R2 = TEMPLATE ADDRESS
                                        ;               $OPRND = PARSED OPTIONS
                                        ;
                                        ; OUTPUT        TEMPLATE UPDATED.
                                        ;-

001066                                  SETDST:
001066  032767  000000G 000000G                 BIT     #XS.PRI,$OPRND  ; PRIORITY ?
001074  001403                                  BEQ     10$             ; IF EQ, NO
001076  116762  000000G 000000G                 MOVB    $PRI,D$SPRI(R2) ; UPDATE TEMPLATE
001104  032767  000000G 000000G 10$:            BIT     #XS.OBJ,$OPRND  ; UPDATE OBJECT NUMBER ?
001112  001406                                  BEQ     20$             ; IF EQ, NO
001114  116762  000000G 000000G                 MOVB    $OBJ,D$SOBJ(R2) ; UPDATE TEMPLATE
001122  042762  000000G 000000G                 BIC     #CS.TSK,C.STS(R2); CLEAR PREVIOUS TASK NAME
001130  032767  000000G 000000G 20$:            BIT     #XS.OBI,$OPRND  ; UPDATE OBJECT TASK ID?
001136  001413                                  BEQ     30$             ; IF EQ, NOPE
001140  [illegible]762  000000G 000000G         MOV     $OBJ,D$STSK(R2) ; UPDATE TEMPLATE
001146  01[illegible]  000002G 000002G          MOV     $OBJ+2,D$STSK+2(R2) ; ...
001154  105062  000000G                         CLRB    D$SOBJ(R2)      ; OBJECT NUMBER IS ZERO
001160  052762  000000G 000000G                 BIS     #CS.TSK,C.STS(R2); REMEMBER OPTION
001166  000207                          30$:    RETURN

```

```
                                                                       M 14
SETMOD - DEFINE MODULE   MACRO V05.03b Monday 01-Jul-85 17:29  Page 16-1
Symbol table

ALCDST  002226R        D$ALEN= ****** GX      ERRDST  000150R        XA.NUM= ****** GX      $CLM   = ****** GX
CHKDST  002144R        D$ANUM= ****** GX      ERRLEN  000545R        XSVCHA  001714R        $CLV   = ****** GX
CS.DSA= ****** GX      D$ASHI= ****** GX      ERRNUM  000374R        XS.CLM= ****** GX      $CT    = ****** GX
CS.DSC= ****** GX      D$ASLO= ****** GX      ERROBJ  000217R        XS.CLV= ****** GX      $CUG   = ****** GX
CS.DST= ****** GX      D$CLEN= ****** GX      ERRPSI  000000R        XS.CTM= ****** GX      $DST   = ****** GX
CS.LLC= ****** GX      D$CMLN= ****** GX      ERRXSV  000436R        XS.CUG= ****** GX      $DTE   = ****** GX
CS.MOD= ****** GX      D$CMSK= ****** GX      ERRX29  000474R        XS.DST= ****** GX      $MCI   = ****** GX
CS.MSK= ****** GX      D$CVAL= ****** GX      EX$ERR= 000002         XS.DTE= ****** GX      $OBJ   = ****** GX
CS.NUM= ****** GX      D$CVLN= ****** GX      L$LCTM= ****** GX      XS.KND= ****** GX      $DPRND= ****** GX
CS.PSN= ****** GX      D$FLNX= ****** GX      L$LNAM= ****** GX      XS.MXC= ****** GX      $PRI   = ****** GX
CS.RDT= ****** GX      D$SLEN= ****** GX      P$SPOR= ****** GX      XS.OBI= ****** GX      $PSIPT= ****** GX
CS.SHI= ****** GX      D$SNAM= ****** GX      RQCB    003012R        XS.OBJ= ****** GX      $RQBF  = ****** GX
CS.SLO= ****** GX      D$SOBJ= ****** GX      R$DADR= ****** GX      XS.PRI= ****** GX      $SEXAC  002472RG
CS.TSK= ****** GX      D$SPRI= ****** GX      R$DLEN= ****** GX      XS.SUB= ****** GX      $SEXSV  000634RG
CS.VAL= ****** GX      D$STSK= ****** GX      R$DNAM= ****** GX      X$9CT  = ****** GX     $SEX29  000646RG
CS.X29= ****** GX      D$STYP= ****** GX      R$LENX= ****** GX      X$9MXC= ****** GX      $SUB1  = ****** GX
C.STS  = ****** GX     ERRALL= ****** GX      SETDSA  001170R        X29CHA  002054R        $SUB2  = ****** GX
DA$CUG= ****** GX      ERRCLM  000261R        SETDSC  001402R        $BYTMN= ****** GX      $TMLST= ****** GX
DEFPRI= 000177         ERRCLV  000326R        SETDST  001066R        $BYTXT= ****** GX      $XXAVL= ****** GX
D$ACUG= ****** GX      ERRCON  000103R        TSTPSI  002770R        $CFERR= ****** GX      .CONER= ****** GX
D$ADLN= ****** GX      ERRDNM  000041R        XA.KND= ****** GX

. ABS.  000000     000    (RW,I,GBL,ABS,OVR)
        003040     001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8485 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:24.42
DB2:SETMOD.T57,[132,134]SETMOD/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETMOD
```

```
                                                                       N 14
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29       PAGE 1
```

```
                                                .SBTTL  UPDREM - UPDATE AREA OF REMOTE DEFINITIONS
                                        ;+
                                        ; **-UPDREM-UPDATE AREA OF REMOTE DEFINITIONS
                                        ;
                                        ; INPUT:        R1  = FORMER EXECUTOR AREA NUMBER (HIORD)
                                        ;               R2  = NEW EXECUTOR AREA NUMBER (HIORD)
                                        ;               $TMLSI  = TEMPLATE LISTHEAD
                                        ;
                                        ; OUTPUT:       ALL REMOTE DEFINITIONS WITH THE OLD EXECUTOR
                                        ;               AREA ARE UPDATED TO REFLECT THE NEW EXECUTOR AREA.
                                        ;
                                        ;-

000354                                  UPDREM:
000354  004567  000000G                         JSR     R5,$SAVRG               ; GET SOME REGISTERS
000360  012705  000000G                         MOV     #$TMLST,R5              ; POINT TO TEMPLATE LISTHEAD
000364  005003                                  CLR     R3                      ; INITIALIZE TEMPORARY LISTHEAD
                                        ;
                                        ; LOCATE NODES WHOSE ADDRESSES REQUIRE CHANGES
                                        ;
000366  010504                          10$:    MOV     R5,R4                   ; SAVE ADDRESS OF PREDECESSOR
000370  011505                                  MOV     (R5),R5                 ; POINT TO NEXT TEMPLATE
000372  001430                                  BEQ     30$                     ; IF EQ, END OF LIST
000374  032765  000000G 000000G 20$:    BIT     #CS.REM,C.STS(R5) ; IS THIS ONE A REMOTE DEFINITION ?
000402  001771                                  BEQ     10$                     ; IF EQ, NO - SKIP IT
000404  016546  000010                          MOV     C.RADD(R5),-(SP); PICK UP REMOTE NODE ADDRESS
000410  042716  001777                          BIC     #1777,(SP)              ; ISOLATE AREA
000414  022601                                  CMP     (SP)+,R1                ; SAME AS OLD EXECUTOR AREA ?
000416  001363                                  BNE     10$                     ; IF NE, NO - LEAVE IT ALONE
000420  042765  176000  000010                  BIC     #^C1777,C.RADD(R5) ; ELSE CLEAR REMOTE AREA
000426  050265  000010                          BIS     R2,C.RADD(R5)           ; FOLD IN NEW AREA
                                        ;
                                        ; REMOVE NODE FROM LIST - INSERT IN TEMPORARY LIST
                                        ;
000432  011546                                  MOV     (R5),-(SP)              ; SAVE NEXT TEMPLATE ADDRESS
000434  010315                                  MOV     R3,(R5)                 ; LINK CHANGED TEMPLATE INTO TEMP LIST
000436  010503                                  MOV     R5,R3                   ;   ... AT THE FRONT ...
000440  012605                                  MOV     (SP)+,R5                ; GET ADDRESS OF NEXT IN SCN
000442  010514                                  MOV     R5,(R4)                 ; CLOSE UP LIST
000444  001353                                  BNE     20$                     ; IF NE, NORE
000446  010367  000002G                         MOV     R3,$TMLST+2             ; ELSE FIX BACK POINTER
000452  000745                                  BR      10$                     ; PROCESS REMAINDER OF REMOTE DEFINITIONS
                                        ;
                                        ; FOLD NODES ON TEMPORARY LIST BACK IN, IN ADDRESS ORDER
                                        ;
000454  005703                          30$:    TST     R3                      ; ANYTHING CHANGED ?
000456  001425                                  BEQ     60$                     ; NO
000460  012705  000000G                         MOV     #$TMLST,R5              ; GET START ADDRESS OF TEMPLATES
000464  010504                          40$:    MOV     R5,R4                   ; SAVE ADDRESS OF PREVIOUS
000466  011505                                  MOV     (R5),R5                 ; GET NEXT TEMPLATE IN LIST
000470  001410                                  BEQ     50$                     ; IF EQ, END OF LIST
000472  122765  000000G 000000G                 CMPB    #CS.REM,C.STS(R5) ; IS THIS A REMOTE TEMPLATE ?
000500  001371                                  BNE     40$                     ; IF NE, NO .. KEEP LOOKING
000502  026365  000010  000010                  CMP     C.RADD(R3),C.RADD(R5) ; ELSE, DOES TEMPLATE GO HERE ?
000510  103365                                  BHIS    40$                     ; IF HIS, NO .. KEEP LOOKING

000512  011446                          50$:    MOV     (R4),-(SP)              ; SAVE OLD SUCCESSOR ADDRESS
000514  010314                                  MOV     R3,(R4)                 ; LINK IN NEW REMOTE
```

```
                                                                  M 16
SETNSP     CREATED BY  MACRO  ON 29-JUN-85 AT 05:22      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE               REFERENCES

CD.LEN  = 000022            #4-59
CL.LEN  = 000020            #4-60
CS.DEC  = ******    GX        5-126
CS.FEA  = ******    GX        5-170
CS.MOD  = ******    GX        5-135      5-140      5-145      5-150      5-155      5-160      5-179      5-189
C.CTM     000016            #4-60
C.DELF    000016            #4-59
C.DELW    000017            #4-59
C.DEV     000012            #4-60
C.EXT     000015            #4-60
C.FLG     000006            #4-60
C.INAC    000014            #4-59      *5-144
C.INCT    000012            #4-59      *5-149
C.IPL     000020            #4-59
C.LIN     000011            #4-60
C.LKS     000010            #4-59      *5-134
C.LST     000021            #4-59
C.NAM     000004            #4-60
C.NCTD  = 000013            #4-60
C.NDC     000006            #4-59      *5-139
C.NLT     000014            #4-60
C.NLTD  = 000012            #4-60
C.OUTT    000013            #4-59      *5-154
C.PRI     000010            #4-60
C.PRM   = ******    GX        5-172      5-177     *5-180     *5-183
C.RETF    000015            #4-59      *5-159
C.SEG     000004            #4-59      *5-190
C.STS   = ******    GX        5-126     *5-135     *5-140     *5-145     *5-150     *5-155     *5-160      5-170     *5-179
                             *5-189
ERRDEC    000000 R          #4-100      5-196
ERRFEA    000030 R          #4-101      5-198
ERRSUP    000066 R          #4-102      5-200
EX$ERR  = 000002            #4-99       4-100      4-101      4-102
E2.INA  = ******    GX        5-142
E2.INC  = ******    GX        5-147
E2.OUT  = ******    GX        5-152
E2.RET  = ******    GX        5-157
E2.SEG  = ******    GX        5-187
NF$ACC  = 100000              5-180      5-183
NF.ACC  = 000001              5-177
OE.LNK  = ******    GX        5-132
OE.NCO  = ******    GX        5-137
OE.VFY  = ******    GX        5-162
SW.VFY  = ******    GX        5-181
$CFERR  = ******    GX        5-196      5-198      5-200
$EINA   = ******    GX        5-144
$EINC   = ******    GX        5-149
$EOUT   = ******    GX        5-154
$ERET   = ******    GX        5-159
$ESEG   = ******    GX        5-190
$LINKS  = ******    GX        5-134
$NCOU   = ******    GX        5-139
```

Table of contents

```
     5-   65    .OBNUM - STORE THE OBJECT NUNMBER
     6-   89    .OBNAM - STORE THE OBJECT TASK NAME
     7-  123    .OBCOP - STORE THE NUMBER OF OBJECT TASK COPIES
```

N 2

```
PRSPRO - ACTION ROUTINES FOR PR MACRO V05.03b Saturday 29-Jun-85 05:19  Page 5
.PRNLT - STORE NUMBER OF LINE TABLES SPECIFIED

                                                 .SBTTL  .PRNLT - STORE NUMBER OF LINE TABLES SPECIFIED
                                         ;+
                                         ; *** - .PRNLT - STORE NUMBER OF LINE TABLES SPECIFIED
                                         ;
                                         ; INPUT:
                                         ;       .PNUMB = NUMBER OF LINE TABLE TO BE SET IN PROCESS SPACE
                                         ;
                                         ; OUTPUT:
                                         ;       NUMBER IS VALIDATED AND THE TRANSITION IS ACCEPTED/REJECTED
                                         ;
                                         ;-

 000064                                  .PRNLT::
 000064  116767  000000G 000000G                 MOVB    .PNUMB,$NLT             ; STORE THE LINE COUNT
 000072  026727  000000G 000000G                 CMP     .PNUMB,#$MXNLT          ; VALUE IN RANGE ?
 000100  101405                                  BLOS    10$                     ; IF LOS, YES
 000102  062716  000002                          ADD     #2,(SP)                 ; ELSE, REJECT
 000106  012767  000000G 000000G                 MOV     #ERRILN,$ERROR          ; INVALID NUMBER
 000114  000207                          10$:    RETURN
```

```
                                             .SBTTL  .RNSPA - STORE SERVICE PASSWORD
                                  ;+
                                  ; ** .RNSPA - STORE REMOTE NODE SERVIDE PASSWORD
                                  ;
                                  ; INPUTS:       TPARS CONVENTIONS.
                                  ;
                                  ; OUTPUT:       SERVICE PASSWORD AT $RNSPA IN RAD50.
                                  ;-

 000642                           .RNSPA::
 000642  012702  000000'                  MOV     #SCR,R2           ; POINT TO SCRATCH
 000646  012701  000010                   MOV     #16./2,R1         ; COUNT
 000652  012722  030060           1$:     MOV     #"00,(R2)+        ; INSERT LEADING ZEROES
 000656  077103                           SOB     R1,1$             ; ...
 000660  016701  000000G                  MOV     .PSTCN,R1         ; GET COUNT
 000664  160102                           SUB     R1,R2             ; INSERT PARSED DIGITS HERE
 000666  020127  000020                   CMP     R1,#16.           ; CHECK LENGTH
 000672  101211                           BHI     REJ               ; REJECT
 000674  016700  000000G                  MOV     .PSTPT,R0         ; POINT TO PARSED STRING

 000700  004767  177700           10$:    CALL    HEX               ; HEX DIGIT ?
 000704  103604                           BCS     REJ               ; IF CS, NO
 000706  112022                           MOVB    (R0)+,(R2)+       ; STUFF BYTE INTO SCRATCH AREA
 000710  077105                           SOB     R1,10$            ; ...

 000712  012700  000000'                  MOV     #SCR,R0           ; SET UP SOURCE ADDRESS,
 000716  012701  000020                   MOV     #16.,R1           ; COUNT,
 000722  012702  000000G                  MOV     #$RNSPA,R2        ; DESTINATION
 000726  000704                           BR      STR50X            ; STORE RAD50 VALUE

 000730  062716  000002           REJ1:   ADD     #2,(SP)           ; REJECT TRANSITION
 000734  000207                           RETURN
```

```
                                                .SBTTL  .SYBYT - POOL SINGLE WORD ALLOCATION EXTENSION
                                        ;+
                                        ; *** - .SYBYT - POOL SINGLE WORD ALLOCATION EXTENSION
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = NUMBER OF BYTES FOR SINGLE WORD POOL EXTENSION
                                        ;
                                        ; OUTPUT:
                                        ;       TRANSITION IS ACCEPTED/REJECTED
                                        ;       NUMBER IS STORED IN $BYT
                                        ;
                                        ;-

000124                                  .SYBYT::
000124  016700  000000G                         MOV     .PNUMB,R0               ; GET THE BYTE COUNT
000130  010067  000000G                         MOV     R0,$BYT                 ; STORE IN ADVERTISED PLACE
000134  020027  177700                          CMP     R0,#<1023.*64.>         ; IS NUMBER IN RANGE ?
000140  101405                                  BLOS    10$                     ; IF LOS, YES - OKAY
000142  062716  000002                          ADD     #2,(SP)                 ; ELSE, REJECT TRANSITION
000146  012767  000000G 000000G                 MOV     #ERRILN,$ERROR          ; INVALID NUMBER
000154  000207                          10$:    RETURN
```

```
                                             .SBTTL  LOCAL DATA
                                             .NLIST  BEX

                                    ;****
                                    ; LOCAL DATA
                                    ;****

                                    ;
                                    ; SPECIAL DESTINATIONS WHICH MAY NOT BE PURGED
                                    ;
 000000   115443   115447   003310  SPCDST: .RAD50  /X25/    /X29ACP/
 000006   115447   050712   000000          .RAD50  /X29/    /MCR   /
 000014   000000                            .WORD   0
                                    ;
                                    ; ERROR TEXT
                                    ;
                                            .NLIST  BEX
          000002                            EX$ERR = 2
 000016      002      103      106  ERRNSD: .ASCIZ  <EX$ERR>/CFE -- Destination not in configuration/
 000067      002      103      106  ERRNDT: .ASCIZ  <EX$ERR>/CFE -- DTE not in configuration/
 000130      002      103      106  ERRNSG: .ASCIZ  <EX$ERR>/CFE -- Group not in configuration/
 000173      002      103      106  ERRXSV: .ASCIZ  <EX$ERR>/CFE -- X25-Server not in configuration/
 000243      002      103      106  ERRX29: .ASCIZ  <EX$ERR>/CFE -- X29-Server not in configuration/
 000313      002      103      106  ERRPSI: .ASCIZ  <EX$ERR>/CFE -- No PSI support in system/
 000354      002      103      106  ERRDLM: .ASCIZ  <EX$ERR>/CFE -- DTE with DLM circuits not purged: %5R/
 000432      002      103      106  ERRDST: .ASCIZ  <EX$ERR>/CFE -- Invalid destination sequence/
                                            .EVEN
                                            .LIST BEX
```

```
                                                                                         N  6
PURMOD - PURGE MODULE      MACRO V05.03b Saturday 29-Jun-85 05:20  Page 15-1
Symbol table

CHKDTE  001004R            DEALC   002064R             ERRNSG  000130R            TSTPSI  002042R            $CFERR= ****** GX
CS.CHN= ****** GX          D$ADLN= ****** GX           ERRPSI  000313R            XA.KND= ****** GX          $CUG   = ****** GX
CS.CUG= ****** GX          D$ALEN= ****** GX           ERRXSV  000173R            XP.CTM= ****** GX          $DST   = ****** GX
CS.DSA= ****** GX          D$CLEN= ****** GX           ERRX29  000243R            XP.CUG= ****** GX          $DTE   = ****** GX
CS.DSC= ****** GX          D$CVLN= ****** GX           EX$ERR= 000002             XP.DTE= ****** GX          $EDMSG= ****** GX
CS.DST= ****** GX          D$FLNX= ****** GX           G$LENX= ****** GX          XP.KND= ****** GX          $OPRND= ****** GX
CS.DTE= ****** GX          D$SLEN= ****** GX           L$LCTM= ****** GX          XP.KNG= ****** GX          $PSIPT= ****** GX
CS.LLC= ****** GX          D$SNAM= ****** GX           L$LENX= ****** GX          XSVCTM  001572R            $PUXAC  001710RG
CS.MOD= ****** GX          D$STYP= ****** GX           L$LNAM= ****** GX          XS.CTM= ****** GX          $PUXPR  000500RG
CS.PVC= ****** GX          D$TADR= ****** GX           PURCUG  001120R            XS.KND= ****** GX          $PUXSV  001246RG
CS.RDT= ****** GX          D$TCT = ****** GX           PURDTE  000536R            X$LENX= ****** GX          $PUXS9  001260RG
CS.X29= ****** GX          D$THSH= ****** GX           P$VLEN= ****** GX          X$9CT = ****** GX          $RLBF  = ****** GX
C$HLEN= ****** GX          D$TLEN= ****** GX           P$VOWN= ****** GX          X29CTM  001650R            $SAVVR= ****** GX
C$LENX= ****** GX          ERRDLM  000354R             R$DLEN= ****** GX          $BUFF = ****** GX          $TMLST= ****** GX
C$ULEN= ****** GX          ERRDST  000432R             R$DNAM= ****** GX          $BYTMN= ****** GX          $XXAVL= ****** GX
C$UNAM= ****** GX          ERRNDT  000067R             R$LENX= ****** GX          $BYTXT= ****** GX          .CONER= ****** GX
C.STS = ****** GX          ERRNSD  000016R             SPCDST  000000R

. ABS.  000000     000     (RW,I,GBL,ABS,OVR)
        002132     001     (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8423 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:12.53
SY:PURMOD.V2,[132,134]PURMOD/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]PURMOD
```

```
                                                .SBTTL  PRGSER - PURGE SERVICE BLOCK PARAMETERS
                                        ;+
                                        ; ** PRGSER - PURGE SERVICE BLOCK PARAMETERS
                                        ;
                                        ; INPUT:        R2 = SER$DF TEMPLATE ADDRESS
                                        ;               $OPRND = PARSED OPTIONS
                                        ;
                                        ; OUTPUT:       THE OPTIONS ARE PURGED.
                                        ;
                                        ; REGISTERS:    R3, R4 AND R5 ARE SAVED.
                                        ;-

000432                                  PRGSER:
000432  012700  000460'                         MOV     #STBL,R0        ; POINT TO CONTROL TABLE
000436  012001                          10$:    MOV     (R0)+,R1        ; PICK UP BIT VALUE
000440  001406                                  BEQ     20$             ; IF EQ, DONE
000442  032067  000000G                         BIT     (R0)+,$OPRND    ; WAS OPTION PARSED ?
000446  001773                                  BEQ     10$             ; IF EQ, NO
000450  040162  000004                          BIC     R1,S$FLG(R2)    ; ELSE CLEAR OPTION IN TEMPLATE
000454  000770                                  BR      10$             ; CONTINUE
000456  000207                          20$:    RETURN

000460                                  STBL:
000460  000002  000000G                         .WORD   SF.HST, RN.HOS                  ; Host address specified
000464  000004  000000G                         .WORD   SF.HAD, RN.HAR                  ; Hardware address specified
000470  000010  000000G                         .WORD   SF.CIR, RN.SCI                  ; Service circuit specified
000474  000020  000000G                         .WORD   SF.DEV, RN.SDV                  ; Service device specified
000500  000040  000000G                         .WORD   SF.PSS, RN.SPA                  ; Service password specified
000504  000100  000000G                         .WORD   SF.DPA, RN.DPA                  ; Dump address specified
000510  000200  000000G                         .WORD   SF.DPC, RN.DCT                  ; Dump count specified
000514  000400  000000G                         .WORD   SF.PH3, RN.SNV                  ; Service node type is Phase 3
000520  000000                                  .WORD   0
```

```
                                        .SBTTL  $SEBUF - SETUP BUFFER TEMPLATE
                                ;+
                                ; *** - $SEBUF - SETUP BUFFER TEMPLATE
                                ;
                                ; INPUT:
                                ;       R3-R5   - TPARS REGISTERS
                                ;
                                ; OUTPUT:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;
                                ;-

000162                          $SEBUF::
000162  010546                          MOV     R5,-(SP)          ; SAVE REGISTER FOR TPARS
000164  012705  000000G                 MOV     #$TMLST,R5        ; POINT AT START OF TEMPLATES
000170  011505                  10$:    MOV     (R5),R5           ; GET NEXT TEMPLATE
000172  001517                          BEQ     101$              ; IF EQ, END OF TEMPLATES
000174  122765  000000G 000000G         CMPB    #CS.BUF,C.STS(R5) ; IS THIS A BUFFER TEMPLATE ?
000202  001372                          BNE     10$               ; IF NE, NO .. KEEP LOOKING
                                ;
                                ; UPDATE THE TEMPLATE
                                ;
000204  052765  000000G 000000G         BIS     #CS.MOD,C.STS(R5) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000212  032767  000000G 000000G         BIT     #OS.CCB,$OPRND    ; NEW CCB NUMBER ?
000220  001403                          BEQ     15$               ; IF EQ, NO
000222  016765  000000G 000004          MOV     $CCBNM,C.CBNM(R5) ; UPDATE THE CCB NUMBER

000230  032767  000000G 000000G 15$:    BIT     #OS.RBN,$OPRND    ; NEW  RDB NUMBER ?
000236  001430                          BEQ     20$               ; IF EQ, NO
000240  016700  000000G                 MOV     $RDBNM,R0         ; GET NEW RDB NUMBER
000244  010065  000010                  MOV     R0,C.RBNM(R5)     ; UPDATE THE RDB NUMBER
000250  005300                          DEC     R0                ; CALCULATE NEW MAX THRESHOLD VALUE
000252  026500  000020                  CMP     C.THSH(R5),R0     ; OLD THRESHOLD VALUE OK ?
000256  101420                          BLOS    20$               ; IF LOS, YES - LEAVE IT
000260  010065  000020                  MOV     R0,C.THSH(R5)     ; ELSE STUFF IT
000264  010502                          MOV     R5,R2             ; COPY BUFFER TEMPLATE ADDRESS
000266  062702  000020                  ADD     #C.THSH,R2        ; POINT TO THRESHOLD VALUE
000272  012701  000041'                 MOV     #ERRTHS,R1        ; POINT TO FORMAT STRING
000276  012700  000000G                 MOV     #$BUFF,R0         ; POINT TO BUFFER
000302  112120                          MOVB    (R1)+,(R0)+       ; MOVE SEVERITY CODE FIRST,
000304  004767  000000G                 CALL    $EDMSG            ; FORMAT IT
000310                                  ERROR$  #$BUFF            ; PRINT WARNING MESSAGE

000320  032767  000000G 000000G 20$:    BIT     #OS.RBS,$OPRND    ; NEW RDB SIZE ?
000326  001414                          BEQ     30$               ; IF EQ, NO
000330  016700  000000G                 MOV     $RDBSZ,R0         ; GET NEW RDB SIZE
000334  166500  000012                  SUB     C.RBSZ(R5),R0     ; CALCULATE DIFFERENCE IN NUMBER RDBS
000340  016501  000010                  MOV     C.RBNM(R5),R1     ; GET RDB NUMBER
000344  016567  000012  177426          MOV     C.RBSZ(R5),RDBSZ  ; SAVE THE CURRENT RDB SIZE
000352  016765  000000G 000012          MOV     $RDBSZ,C.RBSZ(R5) ; UPDATE THE RDB SIZE

000360  032767  000000G 000000G 30$:    BIT     #OS.SBN,$OPRND    ; NEW SDB NUMBER ?
000366  001411                          BEQ     40$               ; IF EQ, NO
000370  016700  000000G                 MOV     $SDBNM,R0         ; GET NEW SDB NUMBER
000374  166500  000014                  SUB     C.SBNM(R5),R0     ; CALCULATE DIFFERENCE IN NUMBER SDBS
000400  016501  000016                  MOV     C.SBSZ(R5),R1     ; GET SDB SIZE IN BYTES
000404  016765  000000G 000014          MOV     $SDBNM,C.SBNM(R5) ; UPDATE THE SDB NUMBER
```

```
SETCNT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:21     PAGE 1

SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE              REFERENCES

CS.CNT  = ******  GX      5-146      6-208
CS.DDM  = ******  GX      5-132      5-144
CS.MOD  = ******  GX      5-155      6-216
C.CNT     000004         #4-102      5-150
C.CSR     000010         #4-104     *5-165
C.LEN   = 000016         #4-107
C.PRI     000012         #4-105     *5-174     6-210      6-212
C.PRM   = ******  GX      5-136     *6-215
C.STS   = ******  GX      5-132      5-144     5-146     *5-155      6-208     *6-216
C.URM     000014         #4-106     *5-159
C.VEC     000006         #4-103     *5-162
ERRLIS  = ******  GX      5-184
ERRPRI    000000 R       #4-92       5-172
EX$ERR  = 000002         #4-91       4-92
OL.CSR  = ******  GX      5-163
OL.PRI  = ******  GX      5-166
OL.URM  = ******  GX      5-157
OL.VEC  = ******  GX      5-160
UPDDDM    000332 R        5-175     #6-201
WC.CNT  = ******  GX      5-148
WC.NAM  = ******  GX      5-134      5-148
$CFERR  = ******  GX      5-172      5-184
$CNTNM  = ******  GX      5-150
$CSR    = ******  GX      5-165
$DVNAM  = ******  GX      5-136
$OPRND  = ******  GX      5-157      5-160     5-163      5-166
$PRI    = ******  GX      5-168      5-170     5-174
$SAVVR  = ******  GX      6-202
$SECNT    000040 RG      #5-124
$TMLST  = ******  GX      5-125
$URM    = ******  GX      5-159
$VEC    = ******  GX      5-162
$WILD   = ******  GX      5-134      5-148
```

N 10

```
**FILE**ID**SETDTE




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
SS           EE               TT       DD      DD       TT       EE
SS           EE               TT       DD      DD       TT       EE
SS           EE               TT       DD      DD       TT       EE
SS           EE               TT       DD      DD       TT       EE
  SSSSSS     EEEEEEEE         TT       DD      DD       TT       EEEEEEEE
  SSSSSS     EEEEEEEE         TT       DD      DD       TT       EEEEEEEE
        SS   EE               TT       DD      DD       TT       EE
        SS   EE               TT       DD      DD       TT       EE
        SS   EE               TT       DD      DD       TT       EE                ....
        SS   EE               TT       DD      DD       TT       EE                ....
SSSSSSSS     EEEEEEEEEE       TT       DDDDDDDD         TT       EEEEEEEEEE         ....
SSSSSSSS     EEEEEEEEEE       TT       DDDDDDDD         TT       EEEEEEEEEE         ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                  N 11
SETDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:21      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00
SYMBOL   VALUE              REFERENCES

CH$HI    = ******  GX        9-318
CS.CHN   = ******  GX        5-88         9-287       9-314
CS.CUG   = ******  GX        5-88
CS.DTE   = ******  GX        5-87         6-119       6-145       10-348
CS.PSN   = ******  GX        5-87
CS.PVC   = ******  GX        5-88
CS.SVC   = ******  GX        5-88
CS.TRB   = ******  GX        8-252        8-258
CS.UNI   = ******  GX        8-252        8-255
CS.X2P   = ******  GX        5-87
CS.X29   = ******  GX        5-87
CS.X3P   = ******  GX        5-87
C$HHI    = ******  GX       *9-319
C$HLEN   = ******  GX        9-292        9-306
C$HLO    = ******  GX       *9-315       *9-317
C.STS    = ******  GX        6-119       *6-145       7-215      *8-252      *8-255      *8-258       9-287      *9-314      *9-318
                             10-348
DEFHSH   = 000040           #4-80         6-143       6-144       6-148
DSTATE     001056 R          6-173      #10-342
D$TADR   = ******  GX        6-125        6-155       10-363
D$TCT    = ******  GX       *6-146       *6-161
D$THSH   = ******  GX       *6-148        6-165      *6-170
D$TLEN   = ******  GX        6-140
D$TLIN   = ******  GX        8-247       10-355       10-357
D$TSTA   = ******  GX       *6-147       10-352      *10-371     *10-374
ERRALL   = ******  GX        6-187
ERRLIN     000052 R         #5-93        10-364
ERRPMI     000012 R         #5-92         6-189
EX$ERR   = 000002           #5-91         5-92
EX$WAR   = 000000           #5-90         5-93
FL.MUX   = ******  GX        8-253
FL.TRB   = ******  GX        8-256
FL.TRI   = ******  GX        8-256
INSDTE     000542 R          6-150       #7-206
L$LENX   = ******  GX        6-143        6-144
PRDLST     000000 R         #5-87         7-214
SETCHN     000674 R          6-179       #9-279
SETLIN     000606 R          6-176       #8-245
XP.CHN   = ******  GX        6-177
XP.COU   = ******  GX        6-159
XP.DTE   = ******  GX        6-151
XP.KND   = ******  GX        6-121        6-135       6-131
XP.LIN   = ******  GX        6-137        6-174
XP.MCI   = ******  GX        6-162
XP.STA   = ******  GX        6-171
XP.ST1   = ******  GX        10-343       10-372
$BUFF    = ******  GX        10-365       10-368
$BYTMN   = ******  GX       *6-143       *6-168
$BYTXT   = ******  GX       *6-144       *6-169
$CFERR   = ******  GX        6-187        6-189       10-368
$CHNLS   = ******  GX        9-299
$CT      = ******  GX        6-161
```

```
                                            .SBTTL  PROEVT - PROCESS EVENT MASK

                                    ;+
                                    ;
                                    ; PROEVT - PROCESS EVENT MASK
                                    ;
                                    ; INPUTS:
                                    ;       $EVENT - FILTER MASKS
                                    ;       $CLASS - EVENT CLASS
                                    ;
                                    ; OUTPUTS:
                                    ;       CARRY CLEAR:
                                    ;          VALID EVENT MASK
                                    ;       CARRY SET:
                                    ;          INVALID EVENT MASK FOUND
                                    ;
                                    ;-
000774  004767  000000G             PROEVT: CALL    $SAVAL           ; SAVE ALL REGISTERS
001000  026727  000000G 074000              CMP     $CLASS,#L.CUST   ; CUSTOMER SPECIFIC EVENT?
001006  103026                              BHIS    50$              ; BR IF YES
001010  012705  000000G                     MOV     #$KNEVT,R5       ; POINT TO VALIDITY TABLE
001014  022567  000000G             10$:    CMP     (R5)+,$CLASS     ; MATCH ON EVENT CLASS?
001020  001013                              BNE     30$              ; BR IF NO
001022  012703  000000G                     MOV     #$EVENT,R3       ; POINT TO FILTER MASK
001026  012701  000004                      MOV     #4,R1            ; PERFORM CHECK ON 4 WORDS
001032  012546                      20$:    MOV     (R5)+,-(SP)      ; GET MASK OF VALID BITS
001034  005116                              COM     (SP)             ; MAKE IT MASK OF INVALID BITS
001036  032623                              BIT     (SP)+,(R3)+      ; ANY INVALID BITS PRESENT?
001040  001007                              BNE     40$              ; BR IF YES
001042  005301                              DEC     R1               ; MORE TO CHECK?
001044  003372                              BGT     20$              ; BR IF YES
001046  000406                              BR      50$              ; ELSE SUCCESSFUL
001050  062705  000010              30$:    ADD     #10,R5           ; POINT TO NEXT ENTRY IN TABLE
001054  005715                              TST     (R5)             ; SCAN ALL OF TABLE
001056  100356                              BPL     10$              ; ...
001060  000261                      40$:    SEC                      ; INDICATE ERROR
001062  000401                              BR      60$
001064  000241                      50$:    CLC                      ; INDICATE SUCCESS
001066  000207                      60$:    RETURN

        000001                              .END
```

```
                                                  .SBTTL  SETDSA - SET DSA$DF PARAMETERS
                                          ;+
                                          ;** SETDSA - SET DSA$DF PARAMETERS
                                          ;
                                          ; INPUTS          R2 = TEMPLATE ADDRESS
                                          ;                 $OPRND = OPTIONS SELECTED
                                          ;
                                          ; OUTPUT          DSA$DF TEMPLATE UPDATED.
                                          ;-

001170                                    SETDSA:
001170  032767  000000G 000000G                   BIT     #XS.CUG,$OPRND  ; CHECK FOR NEW GROUP
001176  001411                                    BEQ     10$             ; IF EQ, NONE
001200  016762  000000G 000000G                   MOV     $CUG,D$ACUG(R2) ; UPDATE TEMPLATE
001206  016762  000002G 000002G                   MOV     $CUG+2,D$ACUG+2(R2);...
001214  052762  000000G 000000G                   BIS     #DA$CUG,C.STS(R2); SET OPTION
001222  032767  000000G 000000G   10$:            BIT     #XS.DTE,$OPRND  ; NEW DTE ADDRESS?
001230  001435                                    BEQ     20$             ; IF EQ, NOPE
001232  016700  177776G                           MOV     $DTE-2,R0       ; GET PARSED LENGTH
001236  006200                                    ASR     R0              ; GET BCD LENGTH
001240  010001                                    MOV     R0,R1           ; COPY IT
001242  166200  000000G                           SUB     D$ADLN(R2),R0   ; GET DIFFERENCE
001246  062700  000003                            ADD     #3,R0           ; ROUND UP TO 4 BYTE BOUND
001252  042700  000003                            BIC     #3,R0           ; ...
001256  060067  000000G                           ADD     R0,$BYTMN       ; UPDATE MINIMUM ALLOCATION
001262  060067  000000G                           ADD     R0,$BYTXT       ; UPDATE ACTUAL ALLOCATION
001266  010162  000000G                           MOV     R1,D$ADLN(R2)   ; UPDATE SAVED LENGTH
001272  012700  000000G                           MOV     #$DTE,R0        ; POINT TO PARSED DTE STRING
001276  010201                                    MOV     R2,R1           ; COPY ADDRESS
001300  062701  000000G                           ADD     #D$ANUM,R1      ; POINT TO NAME FIELD
        000005                                    .REPT   5
                                                  MOV     (R0)+,(R1)+     ; MOVE IT IN
                                                  .ENDR
001316  052762  000000G 000000G                   BIS     #CS.NUM,C.STS(R2) ; SET OPTION
001324  032767  000000G 000000G   20$:            BIT     #XS.SUB,$OPRND  ; SUBADDRESS?
001332  001422                                    BEQ     30$             ; IF EQ, NOPE
001334  042762  000000G 000000G                   BIC     #CS.SLO!CS.SHI,C.STS(R2) ; CLEAR BOTH
001342  052762  000000G 000000G                   BIS     #CS.SLO,C.STS(R2) ; ASSUME ONLY LOW ON COMMAND LINF
001350  016762  000000G 000000G                   MOV     $SUB1,D$ASLO(R2) ; SET LOW RANGE
001356  005767  000000G                           TST     $SUB2           ; TEST FOR PRESENCE OF HIGH END
001362  100406                                    BMI     30$             ; IF MI, NO HIGH END
001364  052762  000000G 000000G                   BIS     #CS.SHI,C.STS(R2) ; REMEMBER OPTION
001372  016762  000000G 000000G                   MOV     $SUB2,D$ASHI(R2); UPDATE TEMPLATE
001400  000207                    30$:            RETURN
```

```
                                                                  N 14
SETMOD      CREATED BY  MACRO  ON 1-JUL-85 AT 17:29        PAGE 1
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL  VALUE              REFERENCES

ALCDST    002226 R          6-161    #13-450
CHKDST    002144 R          6-145    #12-413
CS.DSA  = ******  GX        6-169     13-475
CS.DSC  = ******  GX        6-174     12-420     12-424     12-426     13-480
CS.DST  = ******  GX       12-418     13-471
CS.LLC  = ******  GX       10-356
CS.MOD  = ******  GX       10-363
CS.MSK  = ******  GX        9-293      9-300
CS.NUM  = ******  GX        8-250
CS.PSN  = ******  GX       12-422     14-518
CS.RDT  = ******  GX       14-520     14-545
CS.SHI  = ******  GX        8-253      8-258
CS.SLO  = ******  GX        8-253      8-254
CS.TSK  = ******  GX        7-207      7-213
CS.VAL  = ******  GX        9-279      9-321
CS.X29  = ******  GX       11-384
C.STS   = ******  GX        6-169      6-174     *7-207     *7-213     *8-232     *8-250     *8-253     *8-254     *8-258
                            9-279     *9-293      9-300     *9-321     10-356    *10-363     11-384     12-426     14-518
                           14-520    *14-545
DA$CUG  = ******  GX        8-232
DEFPRI  = 000177           #4-79      13-458
D$ACUG  = ******  GX       *8-230     *8-231
D$ADLN  = ******  GX        8-238     *8-243
D$ALEN  = ******  GX       13-462     13-474
D$ANUM  = ******  GX        8-246
D$ASHI  = ******  GX       *8-259
D$ASLO  = ******  GX       *8-255
D$CLEN  = ******  GX       13-463
D$CMLN  = ******  GX       *9-284      9-302
D$CMSK  = ******  GX        9-287
D$CVAL  = ******  GX        9-315
D$CVLN  = ******  GX        9-281      9-305     *9-312
D$FLNX  = ******  GX       13-467     13-468
D$SLEN  = ******  GX       13-461     13-470
D$SNAM  = ******  GX        6-151      6-153    *13-482    *13-483
D$SOBJ  = ******  GX       *7-206     *7-212
D$SPRI  = ******  GX       *7-203
D$STSK  = ******  GX       *7-210     *7-211
D$STYP  = ******  GX        6-147    *13-484
ERRALL  = ******  GX       13-490     14-566
ERRCLM    000261 R         #5-99       9-330
ERRCLV    000326 R         #5-100      9-328
ERRCON    000103 R         #5-96       6-184
ERRDNM    000041 R         #5-95
ERRDST    000150 R         #5-97      13-492
ERRLEN    000545 R         #5-104      9-326
ERRNUM    000374 R         #5-101     14-569
ERROBJ    000217 R         #5-98      13-494
ERRPSI    000000 R         #5-94      15-585
ERRXSV    000436 R         #5-102     10-368
ERRX29    000474 R         #5-103     11-396
EX$EPR  = 000002           #5-93       5-94       5-95       5-96       5-97       5-98       5-99       5-100      5-101
```

```
    288 000516  012613                              MOV     (SP)+,(R3)      ; POINT TO NEW SUCCESSOR
    289 000520  001355                              BNE     30$             ; IF NE, NO LAST - CONTINUE
    290 000522  010367  000002G                     MOV     R3,$TMLST+2     ; ELSE MARK NEW LIST END
    291 000526  011303                              MOV     (R3),R3         ; POP ADDRESS FROM TEMP LIST
    292 000530  000751                              BR      30$
    293
    294 000532  000207                      60$:    RETURN
    295
    296         000001                              .END
```

```
B  1
C  1
D  1
E  1
F  1
G  1
H  1
I  1
J  1
K  1
L  1
M  1   PRSOBJ
N  1
B  2
C  2
D  2
E  2
F  2
G  2
H  2
I  2
J  2   PRSPRO
K  2
L  2
M  2
N  2
B  3
C  3
D  3
E  3   PRSREM
F  3
G  3
H  3
I  3
J  3
K  3
L  3
M  3
N  3
B  4
C  4
D  4
E  4
F  4
G  4
H  4   PRSSYS
I  4
J  4
K  4
L  4
M  4
N  4
B  5
C  5
D  5
E  5
F  5
G  5
H  5
I  5
```

```
J  5   PURMOD
K  5
L  5
M  5
N  5
B  6
C  6
D  6
E  6
F  6
G  6
H  6
I  6
J  6
K  6
L  6
M  6
N  6
B  7
C  7
D  7
E  7   PURREM
F  7
G  7
H  7
I  7
J  7
K  7
L  7
M  7
N  7
B  8
C  8
D  8
E  8
F  8
G  8
H  8
I  8   SETBUF
J  8
K  8
L  8
M  8
N  8
B  9
C  9
D  9
E  9
F  9   SETCNT
G  9
H  9
I  9
J  9
K  9
L  9
M  9
N  9
B 10
C 10   SETCUG
D 10
```

```
E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10   SETDTE
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12   SETLOG
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13   SETMOD
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14
```

```
M 14
N 14
B 15
C 15
D 15
E 15   SETNOD
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16   SETNSP
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16
```

```
                                                                      B  1
SETNSP      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22     PAGE 2

SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

$OPRND  = ******  GX        5-132      5-137      5-162
$OPRN2  = ******  GX        5-142      5-147      5-152      5-157      5-187
$SENSP    000140 RG        #5-117
$SWICH  = ******  GX        5-181
$TMLST  = ******  GX        5-122      5-167
```

C 1

```
                                                                  B  2
SETOBJ      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE             REFERENCES

CHKMUL    001056 R          5-150      5-191     #7-273
CS.OBJ  = ******   GX       5-143      5-197      6-249
C.CPY     000012           #4-116     *5-169     *5-173     *5-202     *5-211
C.FLG     000005           #4-114     *5-162     *5-165     *5-168     *5-172     *5-176     *5-177     *5-201     *5-205
                           *5-210     *5-214     *5-215      7-311
C.LEN   = 000013           #4-118      5-194
C.NAM     000006           #4-115     *5-158     *5-159     *5-199     *5-200      7-279
C.STS   = ******   GX       5-143     *5-197      6-249
C.TYP     000004           #4-113      5-145     *5-198      6-251      6-251
ERRBUF    000000 R         #4-101      5-226
ERRMUL    000201 R         #4-104      5-224
ERRNAM    000042 R         #4-102      5-228
ERRONA    000115 R         #4-103      5-156      5-186
EX$ERR  = 000002           #4-100      4-101      4-102      4-103      4-104
FINOBJ    001022 R          5-216     #6-244
OBJNAM    000244 R         #4-121      7-283
OF.RLU  = 000100            5-162      5-165      5-205
OF.SMC  = 000200            5-168      5-172      5-210      7-311
OO.COP  = ******   GX       5-166      5-206      7-300
OO.NAM  = ******   GX       5-152      5-184      5-188      7-275
OO.USR  = ******   GX       5-160
OO.VFY  = ******   GX       5-174      5-212
O.LENX  = ******   GX       5-218      5-219
SW.RLU  = ******   GX       5-163      5-203
SW.SMC  = ******   GX       5-170      5-208      7-302
$BYTMN  = ******   GX      *5-218
$BYTXT  = ******   GX      *5-219
$CFERR  = ******   GX       5-156      5-186      5-224      5-226      5-228
$COPES  = ******   GX       5-173      5-211
$C5TA   = ******   GX       7-286      7-288
$OPRND  = ******   GX       5-152      5-160      5-166      5-174      5-184      5-188      5-206      5-212      7-275
                            7-300
$QINSN  = ******   GX       5-217
$RQBF   = ******   GX       5-195
$SAVVR  = ******   GX       7-273
$SEOBJ    000252 RG        #5-136
$SWICH  = ******   GX       5-163      5-170      5-203      5-208      7-302
$TMLST  = ******   GX       5-137      6-245
$TNAM   = ******   GX       5-159      5-159      5-199      5-200      7-274
$TNUM   = ******   GX       5-145      5-154      5-182      5-198
$VLVL   = ******   GX       5-177      5-215
$XXAVL  = ******   GX       5-193
```

C 2

```
                                        .SBTTL  $SEPRO - SET PROCESS PARAMETERS
                                ;+
                                ; *** - $SEPRO - SET PROCESS PARAMETERS
                                ;
                                ; INPUT:
                                ;       $OPRND = OPERANDS PRESENT
                                ;
                                ; OUTPUT:
                                ;       THE PROCESS TEMPLATE IS UPDATED
                                ;
                                ;-

000076                          $SEPRO::
000076  012700  000000G                 MOV     #$TMLST,R0          ; GET START OF TEMPLATE LIST
                                ;
                                ; SEARCH FOR THE PROCESS
                                ;
000102  004767  000000G         10$:    CALL    $FNPRO              ; FIND THE PROCESS TEMPLATE
000106  103467                          BCS     30$                 ; IF CS, NO MORE
                                ;
                                ; UPDATE THE TEMPLATE
                                ;
000110  052760  000000G 000000G         BIS     #CS.MOD,C.STS(R0)   ; INDICATE THAT THIS TEMPLATE IS MODIFIED
000116  032767  000000G 000000G         BIT     #FL.LLC,$FLAGS      ; IS THIS AN LLC PROCESS ?
000124  001423                          BEQ     23$                 ; IF EQ, NO
                                ;
                                ; LLC
                                ;
000126  032767  000000G 000000G         BIT     #OP.NLT,$OPRND      ; LINE TABLE COUNT SPECIFIED ?
000134  001435                          BEQ     25$                 ; IF EQ, NO
000136  026027  000004  114224          CMP     C.NAM(R0),#^RXPT    ; IS THIS PROCESS XPT?
000144  001007                          BNE     21$                 ; IF NE, NO - OKAY
000146  010046                          MOV     R0,-(SP)            ; SAVE R0
000150                                  ERROR$  #ERRNSP             ; XPT CANNOT HAVE LINE TABLES IN PROCESS SPACE
000160  012600                          MOV     (SP)+,R0            ; RESTORE R0
000162  000422                          BR      25$                 ; AND CONTINUE
000164  116760  000000G 000014  21$:    MOVB    $NLT,C.NLT(R0)      ; STORE THE LLC LINE TABLE COUNT
000172  000416                          BR      25$                 ; AND CONTINUE
                                ;
                                ; DDM/DLC
                                ;
000174  032767  000000G 000000G 23$:    BIT     #OP.NLT,$OPRND      ; LINE TABLE COUNT SPECIFIED ?
000202  001403                          BEQ     24$                 ; IF EQ, NO
000204  116760  000000G 000012          MOVB    $NLT,C.NLTD(R0)     ; STORE THE LINE TABLE COUNT
000212  032767  000000G 000000G 24$:    BIT     #OP.NCT,$OPRND      ; CONTROLLER TABLE COUNT SPECIFIED ?
000220  001403                          BEQ     25$                 ; IF EQ, NO
000222  116760  000000G 000013          MOVB    $NCT,C.NCTD(R0)     ; ELSE, STORE THE CONTROLLER TABLE COUNT
                                ;
                                ; PROCESS STATE
                                ;
000230  032767  000000G 000000G 25$:    BIT     #OP.STA,$OPRND      ; DOES HE WANT TO SET THE STATE ?
000236  001721                          BEQ     10$                 ; IF EQ, NO
000240  042760  000000G 000006          BIC     #ZF.MFL,C.FLG(R0)   ; ASSUME MANUAL LOADING
000246  032767  000000G 000000G         BIT     #SW.LOA,$SWICH      ; DOES HE WANT AUTOMATIC LOADING ?
000254  001712                          BEQ     10$                 ; IF EQ, NO
000256  052760  000000G 000006          BIS     #ZF.MFL,C.FLG(R0)   ; MARK THE PROCESS FOR LOAD
000264  000706                          BR      10$                 ; LOOK FOR MORE WORK
```

```
                                         .SBTTL  PRMERR - PRINT PARAMETER ERROR TEXT
                                 ;+
                                 ; ** PRMERR - PRINT PARAMETER ERROR TEXT
                                 ;
                                 ; INPUT:        R0 = PVC TEMPLATE ADDRESS
                                 ;
                                 ; CALLING SEQUENCE:
                                 ;       JSR     R1,PRMERR
                                 ;       .WORD   TEXT ADDRESS, TEXT LENGTH
                                 ;
                                 ; ALL REGISTERS ARE PRESERVED.
                                 ;
                                 ;-

001316                           PRMERR:
001316                                   SAVRG   <R0,R2>                 ; SAVE REGISTERS
001322  012702  000336'                  MOV     #SCR+10,R2              ; POINT TO SCRATCH AREA
001326  005042                           CLR     -(R2)                   ; TRIB NUMBER
001330  156012  000003G                  BISB    P$VNAM+3(R0),(R2)       ;
001334  005042                           CLR     -(R2)                   ; CONTROLLER NUMBER
001336  156012  000002G                  BISB    P$VNAM+2(R0),(R2)       ;
001342  012142                           MOV     (R1)+,-(R2)             ; TEXT STRING ADDRESS
001344  012142                           MOV     (R1)+,-(R2)             ; TEXT SRING LENGTH
001346                                   SAVRG   <R1>                    ; SAVE REGISTER
001350  012700  000000G                  MOV     #$BUFF,R0               ; POINT TO TEXT BUFFER
001354  012701  000210'                  MOV     #ERRPRM,R1              ; POINT TO CONTROL STRING
001360  004767  000000G                  CALL    $EDMSG                  ; BUILD MESSAGE
001364                                   PRINT   #$BUFF                  ; PRINT IT
001374                                   RESRG   <R1,R2,R0>              ; RESTORE REGISTERS
001402  000201                           RTS     R1                      ; RETURN


        000001                           .END
```

```
                                                .SBTTL  FINREM - FIND NEXT REMOTE IN SORTED LIST
                                        ;+
                                        ; *** - FINREM - FIND NEXT REMOTE IN SORTED LIST
                                        ;
                                        ; INPUTS:
                                        ;       R0 = TEMPLATE TO BE INSERTED
                                        ;
                                        ; OUTPUTS:
                                        ;       R1 = ADDRESS OF TEMPLATE TO PRECEED CURRENT TEMPLATE
                                        ;
                                        ;-

000276                                  FINREM:
000276  012702  000000G                         MOV     #$TMLST,R2              ; GET START ADDRESS OF TEMPLATES
000302  010201                          10$:    MOV     R2,R1                   ; SAVE ADDRESS OF PREVIOUS
000304  011202                                  MOV     (R2),R2                 ; GET NEXT TEMPLATE IN LIST
000306  001410                                  BEQ     20$                     ; IF EQ, END OF LIST
000310  122762  000000G 000000G                 CMPB    #CS.REM,C.STS(R2)       ; IS THIS A REMOTE TEMPLATE ?
000316  001371                                  BNE     10$                     ; IF NE, NO .. KEEP LOOKING
000320  026062  000010  000010                  CMP     C.ADD(R0),C.ADD(R2)     ; ELSE, DOES TEMPLATE GO HERE ?
000326  103365                                  BHIS    10$                     ; IF HIS, NO .. KEEP LOOKING
000330  000207                          20$:    RETURN                          ; ELSE, OKAY

```

```
**FILE**ID**SETROU

  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR      000000    UU      UU
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR      000000    UU      UU
SS          EE              TT      RR      RR  00      00  UU      UU
SS          EE              TT      RR      RR  00      00  UU      UU
SS          EE              TT      RR      RR  00      00  UU      UU
SS          EE              TT      RR      RR  00      00  UU      UU
  SSSSSS    EEEEEEEE        TT      RRRRRRRR    00      00  UU      UU
  SSSSSS    EEEEEEEE        TT      RRRRRRRR    00      00  UU      UU
        SS  EE              TT      RR  RR      00      00  UU      UU
        SS  EE              TT      RR  RR      00      00  UU      UU
        SS  EE              TT      RR    RR    00      00  UU      UU      ....
        SS  EE              TT      RR    RR    00      00  UU      UU      ....
SSSSSSSS    EEEEEEEEEE      TT      RR      RR    000000    UUUUUUUUUU      ....
SSSSSSSS    EEEEEEEEEE      TT      RR      RR    000000    UUUUUUUUUU      ....


LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
                                            .MACRO  PRINT   TEXT
                                            MOV     TEXT,R0                 ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                            CALL    $PRINT                  ; PRINT MESSAGE
                                            .ENDM   PRINT

                                            .MACRO  ERROR$  TEXT
                                    ;
                                    ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                    ; MESSAGE STRING.
                                    ;
                                            .IF DIF <TEXT><R0>
                                            MOV     TEXT,R0                         ; GET ADDRESS OF ERROR MESSAGE
                                            .ENDC
                                            CALL    $CFERR                          ; PRINT ERROR MESSAGE
                                            .ENDM   ERROR$

                                            .MACRO  EPRINT  TEXT
                                    ;
                                    ; PRINT TEXT ON ERROR LUN
                                    ;
                                            .IF DIF <TEXT><R0>
                                            MOV     TEXT,R0                         ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                            .ENDC
                                            CALL    $CFER1                          ; PRINT MESSAGE ON ERROR LUN
                                            .ENDM   EPRINT

                                    ;****
                                    ; LOCAL DATA
                                    ;****

                                    ;
                                    ; DEFINE THE SLT TEMPLATE
                                    ;
 000000                                     .ASECT
          000000                            .=0
 000000                                     .BLKW   1               ; C.LNK
 000002                                     .BLKW   1               ; C.STS
 000004                             C.NAM:  .BLKW   1               ; THE DDM PROCESS NAME
 000006                             C.FLG:  .BLKW   1               ; THE SLT FLAGS WORD
 000010                             C.CNT:  .BLKB   1               ; THE DDM CONTROLLER NUMBER
 000011                             C.UNT:  .BLKB   1               ; THE DDM UNIT NUMBER
 000012                             C.CT:   .BLKW   1               ; THE COUNTER TIMER

          000014                    C.LEN=.
 000000                                     .PSECT

 000000                             ARGLST: .BLKW   3

                                    ;
                                    ; ERROR TEXT STRINGS
                                    ;
          000002                            EX$ERR = 2
                                            .NLIST  BEX
 000006      002     103     106    ERRACT: .ASCII  <EX$ERR>/CFE -- Circuits still active, line %R-%M/
 000057      055     045     115    ACTOPT: .ASCIZ  /-%M/
 000063      002     103     106    ERRSTA: .ASCIZ  <EX$ERR>/CFE -- Can't enable multiple circuits on endnode/
                                            .EVEN
```

```
                                                                    B  8
SETSLT - SET SLT PARAMETERS     MACRO V05.03b Saturday 29-Jun-85 05:24  Page 11-2
Symbol table

ACTERR  001226R         EX$ERR= 000002          OC.STA= ****** GX       SW.SDI= ****** GX       $COMP  = ****** GX
ACTOPT  000057R         FL.MTP= ****** GX       OL.CTM= ****** GX       SW.SEN= ****** GX       $CT    = ****** GX
ARGLST  000000R         FL.MUX= ****** GX       OL.STA= ****** GX       S$LBRA= ****** GX       $EDMSG= ****** GX
CHKSTA  001334R         FL.PVC= ****** GX       SETCIR  000504R         S$LCTT= ****** GX       $FLAGS= ****** GX
CO.LIN= ****** GX       FL.TRI= ****** GX       SETLIN  000354R         S$LFLG= ****** GX       $FNSTA= ****** GX
CS.MOD= ****** GX       FNDACT  001160R         SETPTP  000570R         S$LHTM= ****** GX       $HTM   = ****** GX
CS.STA= ****** GX       LF.ENA= ****** GX       SETSTA  001002R         S$LRPR= ****** GX       $OPRND= ****** GX
C.CNT   000010          LF.MFL= ****** GX       SF.ENA= ****** GX       S$LTYP= ****** GX       $SAVRG= ****** GX
C.CT    000012          LF.MTP= ****** GX       SF.SER= ****** GX       S$TFLG= ****** GX       $SAVVR= ****** GX
C.FLG   000006          LF.SER= ****** GX       SL$CTM= ****** GX       WC.CIR= ****** GX       $SESLT  000146RG
C.LEN = 000014          NF.END= ****** GX       SL$HTM= ****** GX       $BUFF  = ****** GX      $STAN  = ****** GX
C.NAM   000004          OC.CTM= ****** GX       SL$TYP= ****** GX       $CFERR= ****** GX       $SWICH= ****** GX
C.STS = ****** GX       OC.HTM= ****** GX       ST$HTM= ****** GX       $CIMXR= ****** GX       $TMLST= ****** GX
C.UNT   000011          OC.MXR= ****** GX       SW.ENA= ****** GX       $CIRPR= ****** GX       $WILD  = ****** GX
ERRACT  000006R         OC.RPR= ****** GX       SW.LOA= ****** GX       $CNFIG= ****** GX       .FNSLT= ****** GX
ERRSTA  000063R

. ABS.  000014    000   (RW,I,GBL,ABS,OVR)
        001572    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8549 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:11.35
SY:SETSLT.V2,[132,134]SETSLT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETSLT
```

C 8

```
SETSVC     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24      PAGE 2
MACRO CROSS REFERENCE                                    CREF    04.00

MACRO NAME       REFERENCES

EPRINT           #4-73
ERROR$           #4-62        5-196        5-213        5-216
PRINT            #4-57
SVCDF$           #4-83        4-85
```

```
SETUNT     CREATED BY  MACRO  ON 12-AUG-85 AT 15:02     PAGE 3

MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

CHADF$           #4-89       4-91
EPRINT           #4-75
ERROR$           #4-64       5-227       7-318
PRINT            #4-59
```

```
SETX3P - SET X.25 LEVEL 3 PARAM MACRO V05.03b Saturday 29-Jun-85 05:25  Page 5
LOCAL DATA

     86                                         .SBTTL  LOCAL DATA
     87                                 ;
     88                                 ; ERROR TEXT
     89                                 ;
     90          000002                         EX$ERR = 2
     91                                         .NLIST  BEX
     92 000000      002      103      106  ERRX3P: .ASCIZ  <EX$ERR>/CFE -- No X3P$DF in database/
     93                                         .LIST   BEX
     94
```

```
                                                                          B 12
SHOBJ - LIST OBJECT CHARACTERIS MACRO V05.03b Saturday 29-Jun-85 05:25  Page 5-2
Symbol table

BUFR     000362R       HEADR    000044R       OFFL   = 000003         O.MXC    000004         WHATL  = 000001
CS.OBJ= ****** GX      HEADR1   000111R       OF.PRO= 000040          O.NAM    000006         $BUFF  = ****** GX
C.CPY    000012        HFLAG    000336R       OF.RLU= 000100          O.TYP    000002         $CBDMG= ****** GX
C.FLG    000005        INSP     000325R       OF.SMC= 000200          O1MSG    000165R        $CFERR= ****** GX
C.LEN  = 000013        INSPL  = 000007        ON        000320R       O2MSG    000202R        $EDMSG= ****** GX
C.NAM    000006        LOGIN    000304R       ONE       000276R       O3MSG    000243R        $GTBUF= ****** GX
C.STS  = ****** GX     LOGINL= 000005         ONEL   = 000006         TBL1     000372R        $PRINT= ****** GX
C.TYP    000004        MS1BLK   000340R       ONL    = 000002         TBL2     000402R        $SHOBJ   000450RG
DEFALT   000311R       MS2BLK   000342R       O.FLG     000003        VTBL     000410R        $TMLST= ****** GX
DFALTL= 000007         MS3BLK   000352R       O.LEN     000012        WC.OBJ= ****** GX       $TNUM  = ****** GX
ERROBJ   000000R       OFF      000322R       O.LNK     000000        WHAT     000334R        $WILD  = ****** GX
EX$ERR= 000002

. ABS.   000013    000   (RW,I,GBL,ABS,OVR)
         001066    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8588 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.04
SY:SHOBJ.V2,[132,134]SHOBJ/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SHOBJ
```

```
                                                                          C 12
SHOBJ        CREATED BY  MACRO  ON 29-JUN-85 AT 05:25        PAGE 1
```

```
    414 002706  004767  000000G                 CALL    $EDMSG          ; FORMAT IT
    415 002712                                  PRINT   #$BUFF          ; PRINT IT
    416 002722  000207                 10$:     RETURN                  ; DONE
    417
```

```
                                                                                  B 14
SHOCIR - LIST CIRCUIT CHARACTER MACRO V05.03b Thursday 29-Aug-85 11:51  Page 13-2
Symbol table

DB2:SHOCIR.T63,[132,134]SHOCIR/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SHOCIR
```

```
                                                                                  C 14
SHOCIR      CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 1
```

```
001642  000167  001052                   JMP     260$               ; ELSE, END OF TEMPLATES

001646  122763  000000G 000000G 80$:     CMPB    #CS.LLC,C.STS(R3)  ; IS THIS AN LLC TEMPLATE ?
001654  001370                           BNE     70$                ; IF NE, NO - KEEP LOOKING
001656  026327  000000G 017704           CMP     C.PRM(R3),#^RECL   ; IS THIS THE ECL LLC ?
001664  001364                           BNE     70$                ; IF NE, NO .. KEEP LOOKING
001666  012703  000000G                  MOV     #$TMLST.,R3        ; POINT AGAIN
                                ;
                                ; SEARCH FOR THE NSP FEATURES TEMPLATE
                                ;
001672  011303                  90$:     MOV     (R3),R3            ; GET NEXT TEMPLATE
001674  001410                           BEQ     100$               ; IF EQ, END OF TEMPLATES
001676  122763  000000G 000000G          CMPB    #CS.FEA,C.STS(R3)  ; IS THIS A FEATURES TEMPLATE ?
001704  001372                           BNE     90$                ; IF NE, NO .. KEEP LOOKING
001706  022763  051516  000000G          CMP     #'NS,C.PRM(R3)     ; IS THIS THE NSP FEATURES TEMPLATE ?
001714  001366                           BNE     90$                ; IF NE, NO .. KEEP LOOKING
                                ;
                                ; SEARCH FOR THE DEC$DF TEMPLATE
                                ;
001716  012704  000000G         100$:    MOV     #$TMLST.,R4        ; POINT AGAIN
001722  011404                  110$:    MOV     (R4),R4            ; GET NEXT TEMPLATE
001724  001002                           BNE     120$               ; BR IF ONE EXISTS
001726  000167  001000                   JMP     270$               ; BR IF NO DEC$DF MACRO
001732  122764  000000G 000000G 120$:    CMPB    #CS.DEC,C.STS(R4)  ; IS THIS THE DEC$DF MACRO?
001740  001370                           BNE     110$               ; BR IF NO

001742  012702  001460'                  MOV     #MSGBLK,R2         ; POINT R2 AT ARGUMENT BLOCK
                                ;
                                ; NODE ID
                                ;
001746  016512  000010                   MOV     C.NID(R5),(R2)     ; STORE THE LENGTH OF THE STRING
001752  010501                           MOV     R5,R1              ; COPY NODE TEMPLATE ADDRESS
001754  062701  000012                   ADD     #C.NID+2,R1        ; CALCULATE ADDRESS OF THE ID STRING
001760  010162  000002                   MOV     R1,2(R2)           ; STORE STRING ADDRESS
001764  012701  000264'                  MOV     #IDMSG,R1          ; POINT R1 AT THE FORMAT STRING
001770  004767  000750                   CALL    PRINT              ; PRINT IT
                                ;
                                ; NAME, ADDRESS, AND HOST
                                ;
001774  012702  001460'                  MOV     #MSGBLK,R2         ; POINT AT ARGUMENT BLOCK
002000  016512  000004                   MOV     C.NAM(R5),(R2)     ; STORE THE NODE NAME
002004  016562  000006  000002           MOV     C.NAM+2(R5),2(R2)  ; ...
002012  012700  000000G                  MOV     #$BUFF,R0          ; POINT TO BUFFER
002016  012701  000314'                  MOV     #NAMMSG,R1         ; DITTO FOR FORMAT STRING
002022  004767  000000G                  CALL    $EDMSG             ; FORMAT EXECUTOR NAME
002026  016501  000052                   MOV     C.NUM(R5),R1       ; GET ADDRESS
002032  004567  000734                   JSR     R5,PUTADR          ; PUT TWO WORD PHASE 4 ADDRESS
002036  000332' 000364'                  .WORD   ADDMSG,ADDMS2      ; ...
002042  004767  001036                   CALL    PUTHST             ; NOW DO HOST, REMOTE NODE STYLE
002046                                   PRINT   #$BUFF             ; SHIP IT
                                ;
                                ; INCOMING TIMER, OUTGOING TIMER
                                ;
002056  012702  001460'                  MOV     #MSGBLK,R2         ; POINT R2 AT ARGUMENT BLOCK
002062  005012                           CLR     (R2)               ; INITIALIZE ARGUMENT BLOCK
002064  005062  000002                   CLR     2(R2)              ; ...
002070  116412  000012                   MOVB    C.INCT(R4),(R2)    ; STORE INCOMING TIMER
```

Table of contents

```
     5-  285    $SHLIN - SHOW LINE CHARACTERISTICS
     6-  365    DISPLY - DISPLAY LINE CHARACTERISTICS
     7-  420    GETUNT - GET THE UNIT INFORMATION
     8-  522    GETURM - GET UNIBUS RUN MASK
     9-  566    GETSLT - GET THE LOADING INFORMATION CO-ROUTINE
    10-  663    FMTLIN - FORMAT A LINE ID
```

```
SETNSP      CREATED BY  MACRO  ON 29-JUN-85 AT 05:22      PAGE 3
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME        REFERENCES

DECTP$            #4-57       4-59
EPRINT            #4-83
ERROR$            #4-72       5-196       5-198       5-200
LLCTP$            #4-57       4-60
NSFDF$            #4-57       4-61
PRINT             #4-67
```

```
SETOBJ     CREATED BY  MACRO  ON 29-JUN-85 AT 05:22     PAGE 2
MACRO CROSS REFERENCE                                  CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-76
ERROR$            #4-65      5-156      5-186      5-224      5-226      5-228
ISTAT$            #4-90
OBJDF$            #4-90      4-92
PRINT             #4-60
STATE$            #4-90
TRAN$             #4-90
```

```

000266  000207                  30$:    RETURN


        000001                          .END
```

```
SETPVC - SET PVC PARAMETERS     MACRO V05.03b Saturday 29-Jun-85 05:23  Page 7-1
Symbol table

CO.KNC= ****** GX        FL.DLM= ****** GX        P$VNAM= ****** GX        WC.UNT= ****** GX        $MBL   = ****** GX
CS.DTE= ****** GX        FOUND    000324R         P$VOWN= ****** GX        X$LENX= ****** GX        $MWI   = ****** GX
CS.PVC= ****** GX        OC.CHN= ****** GX        P$VPSZ= ****** GX        $BUFF  = ****** GX       $OPRND= ****** GX
C$LENX= ****** GX        OC.CTM= ****** GX        P$VSTA= ****** GX        $BYTMN= ****** GX        $PRINT= ****** GX
C.STS = ****** GX        OC.DTE= ****** GX        P$VWSZ= ****** GX        $BYTXT= ****** GX        $PSIPT= ****** GX
D$TADR= ****** GX        OC.MBL= ****** GX        SCR      000326R         $CFERR= ****** GX        $RLBF  = ****** GX
ERRALL= ****** GX        OC.MWI= ****** GX        SETDTE   001212R         $CHN   = ****** GX       $RQBF  = ****** GX
ERRCHN   000145R         OC.STA= ****** GX        SW.ENA= ****** GX        $CIRID= ****** GX        $SEPVC   000346RG
ERRDTE   000045R         PRMERR   001316R         TXDAT    000272R         $CNTNM= ****** GX        $STAN  = ****** GX
ERRPAR   000106R         PV$PSZ= ****** GX        TXDATL= 000014           $COMP  = ****** GX       $SWICH= ****** GX
ERRPRM   000210R         PV$WSZ= ****** GX        TXWND    000306R         $CT    = ****** GX       $TMLST= ****** GX
ERRPVC   000000R         P$VCT  = ****** GX       TXWNDL= 000016           $DTE   = ****** GX       $WILD  = ****** GX
EX$ERR= 000002           P$VLCN= ****** GX        WC.CNT= ****** GX        $EDMSG= ****** GX        $XXAVL= ****** GX
FL.CRE= ****** GX        P$VLEN= ****** GX        WC.TRI= ****** GX        $FLAGS= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001404    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8656 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.06
SY:SETPVC.V2,[132,134]SETPVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETPVC
```

```
                                                .SBTTL  UPDDLL - UPDATE DLL PARAMETERS
                                        ;+
                                        ; ** - UPDDLL - UPDATE DOWNLINE LOAD DATABASE
                                        ;
                                        ; INPUTS:       $OPRND  = PARSED OPTIONS.
                                        ;               R0      = ADDRESS OF REMOTE BLOCK
                                        ;
                                        ; OUTPUT:       THE DLL DATABASE IS UPDATED.
                                        ;-

000332  032767  000000G 000000G UPDDLL: BIT     #RN.XXX,$OPRND          ; DLL PARAMETERS PRESENT ?
000340  001417                          BEQ     20$                     ; IF EQ, NO DONE
000342  004767  000672                  CALL    GETDLL                  ; POINT TO SER$DF,FIL$DF
000346  012705  000402'                 MOV     #DTBL,R5                ; DISPATCH TABLE

000352  012504                  10$:    MOV     (R5)+,R4                ; GET NEXT ROUTINE ADDRESS
000354  001411                          BEQ     20$                     ; IF EQ, DONE
000356  012500                          MOV     (R5)+,R0                ; SOURCE ADDRESS
000360  012503                          MOV     (R5)+,R3                ; GET TEMPLATE FLAG BIT
000362  012501                          MOV     (R5)+,R1                ; GET DESTINATION OFFSET
000364  012502                          MOV     (R5)+,R2                ; GET COUNT (OR TARGET BIT VALUE)
000366  032567  000000G                 BIT     (R5)+,$OPRND            ; WAS THIS OPTION PARSED ?
000372  001767                          BEQ     10$                     ; IF EQ, NOPE
000374  004714                          CALL    (R4)                    ; ELSE CALL ROUTINE
000376  000765                          BR      10$                     ; CONTINUE

000400  000207                  20$:    RETURN

000402                          DTBL:

000402  000640' 000000G 000002  .WORD SETWOR, $RNHOS, SF.HST,  S$HST,  1,      RN.HOS  ;Host node address
000416  000640' 000000G 000020  .WORD SETWOR, $RNSDV, SF.DEV,  S$DEV,  1,      RN.SDV  ;Service device
000432  000662' 000000G 000000  .WORD SETBIT, $SWICH, 0,       S$FLG,  SF.PH3, RN.SNV  ;Service node version
000446  000640' 000000G 000100  .WORD SETWOR, $RNDPA, SF.DPA,  S$DPA,  2,      RN.DPA  ;Dump address
000462  000640' 000000G 000200  .WORD SETWOR, $RNDCP, SF.DPC,  S$DPC,  1,      RN.DCT  ;Dump count
000476  000640' 000000G 000010  .WORD SETWOR, $RNSCI, SF.CIR,  S$CIR,  4,      RN.SCI  ;Service circuit
000512  000640' 000000G 000004  .WORD SETWOR, $RNHAR, SF.HAD,  S$HAD,  4,      RN.HAR  ;Hardware address
000526  000640' 000000G 000040  .WORD SETWOR, $RNSPA, SF.PSS,  S$PSS,  6,      RN.SPA  ;Service password
000542  000702' 000000G 000000  .WORD SETFIL, $RNSLO, 0,       F$SEC,  0,      RN.SLO  ;Secondary loader file
000556  000702' 000000G 000000  .WORD SETFIL, $RNTLO, 0,       F$TER,  0,      RN.TLO  ;Tertiary loader file
000572  000702' 000000G 000000  .WORD SETFIL, $RNLOD, 0,       F$LD,   0,      RN.LOD  ;Load file
000606  000702' 000000G 000000  .WORD SETFIL, $RNDIA, 0,       F$DIA,  0,      RN.DIA  ;Diagnostics file
000622  000702' 000000G 000000  .WORD SETFIL, $RNDMP, 0,       F$DUM,  0,      RN.DMP  ;Dump file
000636  000000                  .WORD  0                                       ;table stopper
```

Table of contents

5- 114 $SEROU - SET ROUTING PARAMETERS

```
    115                                     .LIST BEX
    116
```

```
                                                                C  8
SETSLT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL    VALUE             REFERENCES

ACTERR     001226 R             5-220   #10-426
ACTOPT     000057 R            #3-112   *10-435   *10-438
ARGLST     000000 R            #3-104    10-431    10-441
CHKSTA     001334 R             4-142   #11-456
CO.LIN  =  ******   GX          4-139     4-174     7-293
CS.MOD  =  ******   GX          4-151     8-357     8-365
CS.STA  =  ******   GX          9-405
C.CNT      000010               #3-97    10-433
C.CT       000012               #3-99
C.FLG      000006               #3-96     7-309
C.LEN   =  000014              #3-101
C.NAM      000004               #3-95    10-432
C.STS   =  ******   GX         *4-151    *4-156     7-311    *7-319    *8-357    *8-365    *8-367     9-405
C.UNT      000011               #3-98    10-439
ERRACT     000006 R            #3-111    10-429
ERRSTA     000063 R            #3-113    11-501
EX$ERR  =  000002              #3-109     3-111     3-113
FL.MTP  =  ******   GX          8-347     9-396    11-487
FL.MUX  =  ******   GX         10-436
FL.PVC  =  ******   GX          4-136
FL.TRI  =  ******   GX          4-141
FNDACT     001160 R             5-218     6-263    #9-395
LF.ENA  =  ******   GX          5-225     6-262     6-265     7-299     7-305    11-491
LF.MFL  =  ******   GX          5-202     5-206     5-225     5-228     6-258     6-265     7-295     7-299     7-302
LF.MTP  =  ******   GX          5-196     6-251     7-309    11-477
LF.SER  =  ******   GX          7-325     7-328
NF.END  =  ******   GX         11-464
OC.CTM  =  ******   GX          4-153
OC.HTM  =  ******   GX          4-172     7-307     8-363
OC.MXR  =  ******   GX          4-158
OC.RPR  =  ******   GX          4-162
OC.STA  =  ******   GX          4-172     7-291     8-355    11-462
OL.CTM  =  ******   GX          4-153
OL.STA  =  ******   GX          4-172     8-355    11-462
SETCIR     000504 R             4-178    #6-247
SETLIN     000354 R             4-176    #5-195
SETPTP     000570 R             5-198     6-253    #7-285
SETSTA     001002 R             5-217     6-257    #8-346
SF.ENA  =  ******   GX          8-358     8-361     9-407    11-498
SF.SER  =  ******   GX          8-371     8-374
SL$CTM  =  ******   GX          4-156
SL$HTM  =  ******   GX          7-319
SL$TYP  =  ******   GX          7-311
ST$HTM  =  ******   GX          8-367
SW.ENA  =  ******   GX          5-213     6-260     7-297     7-303     8-359    11-466
SW.LGA  =  ******   GX          5-204     5-226     7-300
SW.SDI  =  ******   GX          4-170     7-323     7-326     8-369     8-372
SW.SEN  =  ******   GX          4-170     7-323     8-369
S$LBRA  =  ******   GX         *4-160
S$LCTT  =  ******   GX         *4-155
S$LFLG  =  ******   GX          5-196     5-202    *5-206    *5-225    *5-228     6-251     6-258    *6-262    *6-265
                                7-295    *7-299    *7-302    *7-305    *7-325    *7-328    11-491

                                                                D  8
```

```
**FILE**ID**SETUNT






  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   UU      UU   NN      NN   TTTTTTTTTT
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   UU      UU   NN      NN   TTTTTTTTTT
SS           EE               TT       UU      UU   NN      NN       TT
SS           EE               TT       UU      UU   NN      NN       TT
SS           EE               TT       UU      UU   NNNN    NN       TT
SS           EE               TT       UU      UU   NNNN    NN       TT
  SSSSSS     EEEEEEEE         TT       UU      UU   NN  NN  NN       TT
  SSSSSS     EEEEEEEE         TT       UU      UU   NN  NN  NN       TT
        SS   EE               TT       UU      UU   NN    NNNN       TT
        SS   EE               TT       UU      UU   NN    NNNN       TT
        SS   EE               TT       UU      UU   NN      NN       TT            ....
        SS   EE               TT       UU      UU   NN      NN       TT            ....
SSSSSSSS     EEEEEEEEEE       TT       UUUUUUUUUU   NN      NN       TT            ....
SSSSSSSS     EEEEEEEEEE       TT       UUUUUUUUUU   NN      NN       TT            ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

C 10

```
**FILE**ID**SETX2P






  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   XX      XX     222222     PPPPPPPP
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   XX      XX     222222     PPPPPPPP
SS           EE               TT       XX      XX   22      22   PP      PP
SS           EE               TT       XX      XX   22      22   PP      PP
SS           EE               TT         XX  XX             22   PP      PP
SS           EE               TT         XX  XX             22   PP      PP
  SSSSSS     EEEEEEEE         TT           XX             22     PPPPPPPP
  SSSSSS     EEEEEEEE         TT           XX             22     PPPPPPPP
        SS   EE               TT         XX  XX         22       PP
        SS   EE               TT         XX  XX         22       PP
        SS   EE               TT       XX      XX     22         PP               ....
        SS   EE               TT       XX      XX     22         PP               ....
SSSSSSSS     EEEEEEEEEE       TT       XX      XX   2222222222   PP               ....
SSSSSSSS     EEEEEEEEEE       TT       XX      XX   2222222222   PP               ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
     96                                                   .SBTTL  $SEX3P - SET X.25 LEVEL 3 PARAMETERS
     97                                          ;+
     98                                          ; *** - $SEX3P - SET X.25 LEVEL 3 PARAMETERS
     99                                          ;
    100                                          ; INPUT:
    101                                          ;        $OPRND = OPERANDS PRESENT
    102                                          ;
    103                                          ; OUTPUT:
    104                                          ;        C-BIT = SUCCESS/FAILURE
    105                                          ;        THE X3P$DF TEMPLATE IS UPDATED
    106                                          ;
    107                                          ;-
    108
    109 000036                                   $SEX3P::
    110 000036  012700  000000G                          MOV     #$PSIPT,R0       ; POINT TO PSN$DF TEMPLATE
    111 000042  011000                           10$:    MOV     (R0),R0          ; NEXT TEMPLATE
    112 000044  001522                                   BEQ     101$             ; IF EQ, NO MORE
    113 000046  122760  000000G 000000G                  CMPB    #CS.X3P,C.STS(R0) ; IS THIS A X3P$DF TEMPLATE ?
    114 000054  001372                                   BNE     10$              ; IF NE, NO - KEEP LOOKING
    115 000056  032767  000000G 000000G                  BIT     #XP.CTM,$OPRND   ; CALL TIMER?
    116 000064  001403                                   BEQ     20$              ; BR IF NO
    117 000066  116760  000000G 000000G                  MOVB    $CALT,X$3CT(R0)  ; UPDATE TEMPLATE
    118
    119 000074  032767  000000G 000000G 20$:             BIT     #XP.CLT,$OPRND   ; CLEAR TIMER?
    120 000102  001403                                   BEQ     30$              ; BR IF NO
    121 000104  116760  000000G 000000G                  MOVB    $CLT,X$3KT(R0)   ; UPDATE TEMPLATE
    122
    123 000112  032767  000000G 000000G 30$:             BIT     #XP.DFB,$OPRND   ; DEFAULT DATA?
    124 000120  001403                                   BEQ     40$              ; BR IF NO
    125 000122  016760  000000G 000000G                  MOV     $DFD,X$3DFP(R0)  ; UPDATE TEMPLATE
    126
    127 000130  032767  000000G 000000G 40$:             BIT     #XP.DFW,$OPRND   ; DEFAULT WINDOW?
    128 000136  001403                                   BEQ     50$              ; BR IF NO
    129 000140  116760  000000G 000000G                  MOVB    $DFW,X$3DW(R0)   ; UPDATE TEMPLATE
    130
    131 000146  032767  000000G 000000G 50$:             BIT     #XP.MXW,$OPRND   ; MAXIMUM WINDOW?
    132 000154  001403                                   BEQ     55$              ; BR IF NO
    133 000156  116760  000000G 000000G                  MOVB    $MWI,X$3MW(R0)   ; UPDATE DATA
    134
    135 000164  032767  000000G 000000G 55$:             BIT     #XP.MXB,$OPRND   ; MAXIMUM DATA?
    136 000172  001403                                   BEQ     60$              ; BR IF NO
    137 000174  016760  000000G 000000G                  MOV     $BSZ,X$3MXP(R0)  ; UPDATE VALUE
    138
    139 000202  032767  000000G 000000G 60$:             BIT     #XP.MXC,$OPRND   ; MAXIMUM CLEARS?
    140 000210  001403                                   BEQ     65$              ; BR IF NO
    141 000212  116760  000000G 000000G                  MOVB    $MXC,X$3KM(R0)   ; UPDATE TEMPLATE
    142
    143 000220  032767  000000G 000000G 65$:             BIT     #XP.MRE,$OPRND   ; MAXIMUM RESETS?
    144 000226  001403                                   BEQ     70$              ; BR IF NO
    145 000230  116760  000000G 000000G                  MOVB    $MRE,X$3RM(R0)   ; UPDATE VALUE
    146
    147 000236  032767  000000G 000000G 70$:             BIT     #XP.MXR,$OPRND   ; MAXIMUM RESTARTS?
    148 000244  001403                                   BEQ     75$              ; BR IF NO
    149 000246  116760  000000G 000000G                  MOVB    $MXR,X$3SM(R0)   ; UPDATE TEMPLATE
    150
    151 000254  032767  000000G 000000G 75$:             BIT     #XP.RET,$OPRND   ; RESET TIMER?
    152 000262  001403                                   BEQ     80$              ; BR IF NO
```

```
                                                                  C 12
SHOBJ      CREATED BY  MACRO  ON 29-JUN-85 AT 05:25      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE             REFERENCES

BUFR       000362 R         #4-142     5-233      5-237      5-239
CS.OBJ   = ******   GX       5-192
C.CPY      000012           #4-109     5-232
C.FLG      000005           #4-107     5-216      5-224      5-230
C.LEN    = 000013           #4-110
C.NAM      000006           #4-108     5-214      5-215
C.STS    = ******   GX       5-192
C.TYP      000004           #4-106     5-196      5-213
DEFALT     000311 R         #4-126     4-127      5-227
DFALTL   = 000007           #4-127     5-226
ERROBJ     000000 R         #4-116     5-260
EX$ERR   = 000002           #4-115     4-116
HEADR      000044 R         #4-117     5-204
HEADR1     000111 R         #4-118     5-207
HFLAG      000336 R         #4-138    *5-185      5-201     *5-203
INSP       000325 R         #4-132     4-133      4-160
INSPL    = 000007           #4-133     4-160
LOGIN      000304 R         #4-124     4-125      5-223
LOGINL   = 000005           #4-125     5-222
MS1BLK     000340 R         #4-139     4-153     *5-212     *5-213
MS2BLK     000342 R         #4-140     4-154     *5-214     *5-215     *5-220     *5-221
MS3BLK     000352 R         #4-141     4-155     *5-222     *5-223     *5-226     *5-227     *5-228     *5-229     *5-238
                            *5-239
OFF        000322 R         #4-130     4-131      4-161
OFFL     = 000003           #4-131     4-161
OF.RLU   = 000100            5-224
OF.SMC   = 000200            5-230
ON         000320 R         #4-128     4-129      4-159
ONE        000276 R         #4-122     4-123      5-229
ONEL     = 000006           #4-123     5-228
ONL      = 000002           #4-129     4-159
O1MSG      000165 R         #4-119     4-146
O2MSG      000202 R         #4-120     4-147
O3MSG      000243 R         #4-121     4-148
TBL1       000372 R         #4-146     5-240
TBL2       000402 R         #4-153     5-241
VTBL       000410 R         #4-159     5-220      5-221
WC.OBJ   = ******   GX       5-194     5-205      5-254
WHAT       000334 R         #4-134     4-135      4-162      4-163      4-164      4-165      4-166
WHATL    = 000001           #4-135     4-162      4-163      4-164      4-165      4-166
$BUFF    = ******   GX       5-208     5-211      5-245      5-247
$CBDMG   = ******   GX       5-235
$CFERR   = ******   GX       5-260
$EDMSG   = ******   GX       5-210     5-246
$GTBUF   = ******   GX       5-209
$PRINT   = ******   GX       5-211     5-247
$SHOBJ     000450 RG        #5-183
$TMLST   = ******   GX       5-186
$TNUM    = ******   GX       5-196
$WILD    = ******   GX       5-194     5-205      5-254
```

```
                                                .SBTTL  GETTCI - GET TRANSPORT CIRCUIT INFORMATION
                                        ;+
                                        ; *** - GETTCI - GET TRANSPORT CIRCUIT INFORMATION
                                        ;
                                        ; INPUT:
                                        ;       R5 = ADDRESS OF THE CURRENT UNIT DEFINITION
                                        ;
                                        ; OUTPUT:
                                        ;       CARRY SET: NO TRANSPORT CIRCUITS FOR THIS LINE
                                        ;       CARRY CLEAR:
                                        ;               TYPBLK+0,+2     PROTOCOL TYPE
                                        ;               TYPBLK+4,+6     OWNER NAME
                                        ;               HELBLK+0        HELLO TIMER (PTP)
                                        ;               HELBLK+2,+4     SERVICE (PTP)
                                        ;               STABLK+0        COST FROM UNI$DF
                                        ;
                                        ;-

002724                                  GETTCI:
                                        ;
                                        ; PROTOCOL TYPE
                                        ;
002724  012700  001544'                 10$:    MOV     #HWP,R0                 ; POINT TO DDM NAME BUFFER
002730  016701  000000C                         MOV     $DVNAM+L$NAM,R1         ; GET DEVICE NAME
002734  004767  000000G                         CALL    $C5TA                   ; CONVERT IN CASE HARDWARE PROTOCOL
002740  005004                                  CLR     R4                      ; ASSUME LAPB PROTOCOL
002742  016500  000010                          MOV     C.CW1(R5),R0            ; GET CHARACTERISTICS
002746  042700  177770                          BIC     #^C7,R0                 ; ISOLATE PROTOCOL TYPE
002752  006300                                  ASL     R0                      ; MAKE IT A
002754  006300                                  ASL     R0                      ;  DOUBLEWORD INDEX
002756  016067  001550' 176756                  MOV     PROTBL+0(R0),TYPBLK+4 ; GET LENGTH
002764  016067  001552' 176752                  MOV     PROTBL+2(R0),TYPBLK+6 ; GET ADDRESS
002772  001033                                  BNE     25$                     ; IF NE, NOT DDCMP
002774  005000                                  CLR     R0                      ; ASSUME POINT TO POINT
002776  042767  000000G 000000G                 BIC     #FL.SLV,$FLAGS          ; NOT A MULTIPOINT TRIB
003004  032765  000010  000010                  BIT     #CH.TYP,C.CW1(R5) ; CHECK FOR MULTIPOINT
003012  001413                                  BEQ     20$                     ; IF EQ, POINT TO POINT
003014  005200                                  INC     R0                      ; ELSE PROTOCOL = DDCMP CONTROL
003016  032765  000020  000010                  BIT     #CH.MDT,C.CW1(R5) ; TRIBUTARY ?
003024  001406                                  BEQ     20$                     ; IF EQ, PROTOCOL = DDCMP CONTROL
003026  005200                                  INC     R0                      ; ELSE PROTOCOL = DDCMP TRIBUTARY
003030  052767  000000C 000000G                 BIS     #FL.SLV!FL.TRI,$FLAGS   ; REMEMBER FOR LINE FORMATTING LATER
003036  005067  000000G                         CLR     $STAN                   ; MATCH FIRST (ONLY) STATION
003042  006300                          20$:    ASL     R0                      ; MAKE IT A DOUBLEWORD
003044  006300                                  ASL     R0                      ;  INDEX
003046  016067  001610' 176666                  MOV     DDCTBL+0(R0),TYPBLK+4 ; STRING LENGTH
003054  016067  001612' 176662                  MOV     DDCTBL+2(R0),TYPBLK+6 ; STRING LENGTH

003062  012767  000010  176646          25$:    MOV     #EXECL,TYPBLK+0 ; ASSUME OWNER = EXECUTOR
003070  012767  001335' 176642                  MOV     #EXEC,TYPBLK+2  ;  ...
                                        ;
                                        ; GET UNIT COST
                                        ;
003076  016567  000014  176724                  MOV     C.CST(R5),STABLK+0 ; COST FOR UNIT
003104  005727                                  TST     (PC)+                   ; RETURN SUCCESS
003106  000261                          30$:    SEC                             ; RETURN CARRY SET IF NO TRANSPORT CIRCUIT
003110  000207                                  RETURN
```

```
                                                                   C 14
SHOCIR      CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE                REFERENCES

APR        001471 R         #4-197      8-555       8-557
APRVAL     001520 R         #4-198      8-552
CH.MDT   = 000020            7-456
CH.TYP   = 000010            7-453
CO.KNC   = ******   GX       5-332      10-644      12-754
CS.CNT   = ******   GX       5-299      5-319
CS.DDM   = ******   GX       5-290
CS.DTE   = ******   GX       5-343      5-348
CS.PVC   = ******   GX       5-350
CS.SVC   = ******   GX       10-638
CS.UNI   = ******   GX       12-853
CS.UNT   = ******   GX       5-315
CS.X3P   = ******   GX       5-340
CURSLT     000000 R         #4-117     *5-293       6-371       6-404       8-500      *8-506
C.CSR      000012           #4-128
C.CST      000014           #4-129      7-471
C.CWO      000006           #4-126
C.CW1      000010           #4-127      7-444       7-453       7-456
C.DPR      000016           #4-130
C.HTM      000020           #4-131
C.LEN    = 000024           #4-133
C.LTM      000022           #4-132
C.PRM    = ******   GX       5-288      5-292       5-306       5-308       5-326       5-328       8-539       8-551       8-559
                             8-561
C.STS    = ******   GX       5-290      5-299       5-315       5-319       5-340       5-343       5-348       5-350       10-638
                             10-669     10-674      10-698      12-836      12-841      12-853
C.UNT      000004           #4-125
DDCTBL     001610 R         #4-212      7-463       7-464
DISPLY     002504 R          5-330     #6-369
DSAB       001427 R         #4-193      4-193       8-513       8-535
DSABL    = 000010           #4-193      8-512       8-534
D$TADR   = ******   GX       12-818
D$TLIN   = ******   GX       12-850
ENAB       001420 R         #4-192      4-192       8-509       8-531
ENABL    = 000007           #4-192      8-508       8-530
ERRCIS   = ******   GX       6-393
ERRNST     001242 R         #4-177
EXEC       001333 R         #4-184      4-184       7-467       12-811
EXECL    = 000010           #4-184      7-466       12-810
EX$ERR   = 000002           #4-138     #4-176       4-177
FL.DLC   = ******   GX       3-549
FL.DLM   = ******   GX       10-646     12-755      12-779
FL.MTP   = ******   GX       6-374      8-519       8-568       9-605
FL.MUX   = ******   GX       5-322      9-599       12-855
FL.SLV   = ******   GX       7-452      7-459       9-607
FL.TRB   = ******   GX       6-383      6-387
FL.TRI   = ******   GX       5-283      5-286       7-459       8-521       8-570
FL.X25   = ******   GX       5-273
FMTDLM     000406 R         #4-155      12-785
FMTLIN     003536 R          6-399     #9-591
FMTLN1     003542 R         #9-592
FMTLN2     003546 R         #9-593      12-856
```

```
                                                                   D 14
SHOCIR      CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 2
```

```
 002074  116462  000013  000002          MOVB    C.OUTT(R4),2(R2); STORE OUTGOING TIMER
 002102  012701  001230'                 MOV     #TIMMSG,R1      ; POINT R1 AT PRINT STRING
 002106  004767  000632                  CALL    PRINT           ; PRINT IT
                                 ;
                                 ; MAXIMUM LINKS, MAXIMUM NODE COUNTERS
                                 ;
 002112  012702  001460'                 MOV     #MSGBLK,R2      ; POINT R2 AT ARGUMENT BLOCK
 002116  016412  000010                  MOV     C.LKS(R4),(R2)  ; GET THE MAX # OF LOG. LINKS
 002122  016462  000006  000002          MOV     C.NDC(R4),2(R2) ; STORE MAXIMUM NODE COUNTERS
 002130  012701  000434'                 MOV     #HOSMSG,R1      ; POINT R1 AT THE INPUT STRING
 002134  004767  000604                  CALL    PRINT           ; PRINT IT
                                 ;
                                 ; DELAY FACTOR, DELAY WEIGHT
                                 ;
 002140  012702  001460'                 MOV     #MSGBLK,R2      ; POINT R2 AT ARGUMENT BLOCK
 002144  005012                          CLR     (R2)            ; INITIALIZE ARGUMENT BLOCK
 002146  005062  000002                  CLR     2(R2)           ; ...
 002152  005001                          CLR     R1              ; GET DELAY FACTOR VALUE
 002154  156401  000016                  BISB    C.DELF(R4),R1   ; ...
 002160  012700  000001                  MOV     #1,R0           ; DELAY FACTOR STORED AS (POWER OF 2)/16.
 002164  062701  000004                  ADD     #4,R1           ; ...
 002170  006300                  130$:   ASL     R0              ; RAISE DELAY FACTOR TO POWER OF 2
 002172  005301                          DEC     R1              ; ...
 002174  003375                          BGT     130$            ; ...
 002176  110012                          MOVB    R0,(R2)         ; STORE DELAY FACTOR
 002200  012700  000001                  MOV     #1,R0           ; DELAY WEIGHT STORED AS POWER OF 2
 002204  005001                          CLR     R1              ; GET DELAY WEIGHT VALUE
 002206  156401  000017                  BISB    C.DELW(R4),R1   ; ...
 002212  001403                          BEQ     145$            ; BR IF DELAY WEIGHT IS 0
 002214  006300                  140$:   ASL     R0              ; RAISE DELAY WEIGHT TO POWER OF 2
 002216  005301                          DEC     R1              ; ...
 002220  003375                          BGT     140$            ; ...
 002222  110062  000002          145$:   MOVB    R0,2(R2)        ; STORE DELAY WEIGHT
 002226  012701  001303'                 MOV     #DELMSG,R1      ; POINT R1 AT THE PRINT STRING
 002232  004767  000506                  CALL    PRINT           ; PRINT IT
                                 ;
                                 ; INACTIVITY TIMER, RETRANSMIT FACTOR
                                 ;
 002236  012702  001460'                 MOV     #MSGBLK,R2      ; POINT R2 AT ARGUMENT BLOCK
 002242  005012                          CLR     (R2)            ; INITIALIZE ARGUMENT BLOCK
 002244  005062  000002                  CLR     2(R2)           ; ...
 002250  116412  000014                  MOVB    C.INAC(R4),(R2) ; STORE INACTIVITY TIMER
 002254  116462  000015  000002          MOVB    C.RETF(R4),2(R2); STORE RETRANSMIT FACTOR
 002262  012701  001352'                 MOV     #RETMSG,R1      ; POINT R1 AT THE PRINT STRING
 002266  004767  000452                  CALL    PRINT           ; PRINT IT
                                 ;
                                 ; TYPE, VERIFICATION STATE AND MAXIMUM ADDRESS
                                 ;
 002272  012702  001460'                 MOV     #MSGBLK,R2      ; POINT AT THE MESSAGE BLOCK
 002276  012700  001442'                 MOV     #ATYP,R0        ; POINT TO AREA STRING ADDRESS
 002302  032767  000100  000000G         BIT     #NF.LV2,$CNFIG  ; LEVEL 2 ROUTER ?
 002310  001006                          BNE     150$            ; IF NE, YES
 002312  022020                          CMP     (R0)+,(R0)+     ; ELSE ASSUME LEVEL 1 ROUTER
 002314  032767  000200  000000G         BIT     #NF.END,$CNFIG  ; BR IF ITS LEVEL 1
 002322  001401                          BEQ     150$            ; IF EQ, NO
 002324  022020                          CMP     (R0)+,(R0)+     ; ELSE ITS NONROUTING
 002326  012022                  150$:   MOV     (R0)+,(R2)+     ; LOAD UP TYPE LENGTH
```

```
        .TITLE  SHOLIN - LIST LINE CHARACTERISTICS
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SHOW LINE CHARACTERISTICS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;



```

```
                                                       D  1
**FILE**ID**SETOBJ




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     000000     BBBBBBBB             JJ
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     000000     BBBBBBBB             JJ
SS           EE               TT       00      00   BB      BB           JJ
SS           EE               TT       00      00   BB      BB           JJ
SS           EE               TT       00      00   BB      BB           JJ
SS           EE               TT       00      00   BB      BB           JJ
  SSSSSS     EEEEEEEE         TT       00      00   BBBBBBBB             JJ
  SSSSSS     EEEEEEEE         TT       00      00   BBBBBBBB             JJ
        SS   EE               TT       00      00   BB      BB   JJ      JJ
        SS   EE               TT       00      00   BB      BB   JJ      JJ
        SS   EE               TT       00      00   BB      BB   JJ      JJ        ....
        SS   EE               TT       00      00   BB      BB   JJ      JJ        ....
SSSSSSSS     EEEEEEEEEE       TT         000000     BBBBBBBB       JJJJJ           ....
SSSSSSSS     EEEEEEEEEE       TT         000000     BBBBBBBB       JJJJJ           ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                       E  1
```

```
**FILE**ID**SETPAR
```

```
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP       AAAAAA     RRRRRRRR
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP       AAAAAA     RRRRRRRR
SS           EE               TT       PP      PP   AA      AA   RR      RR
SS           EE               TT       PP      PP   AA      AA   RR      RR
SS           EE               TT       PP      PP   AA      AA   RR      RR
SS           EE               TT       PP      PP   AA      AA   RR      RR
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     AA      AA   RRRRRRRR
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     AA      AA   RRRRRRRR
        SS   EE               TT       PP           AAAAAAAAAA   RR  RR
        SS   EE               TT       PP           AAAAAAAAAA   RR  RR
        SS   EE               TT       PP           AA      AA   RR    RR         ....
        SS   EE               TT       PP           AA      AA   RR    RR         ....
SSSSSSSS     EEEEEEEEEE       TT       PP           AA      AA   RR      RR       ....
SSSSSSSS     EEEEEEEEEE       TT       PP           AA      AA   RR      RR       ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
SETPRO - SET PROCESS OPTIONS    MACRO V05.03b Saturday 29-Jun-85 05:23  Page 5-2
Symbol table

CL.LEN= 000020          C.LIN   000011          C.STS = ****** GX       OP.STA= ****** GX       $NCT  = ****** GX
CS.MOD= ****** GX       C.NAM   000004          ERRNSP  000000R         SW.LOA= ****** GX       $NLT  = ****** GX
C.CTM   000016          C.NCTD= 000013          EX$ERR= 000002          ZF.MFL= ****** GX       $OPRND= ****** GX
C.DEV   000012          C.NIT   000014          FL.LLC= ****** GX       $CFERR= ****** GX       $SEPRO  000076RG
C.EXT   000015          C.NLID= 000012          OP.NCT= ****** GX       $FLAGS= ****** GX       $SWICH= ****** GX
C.FLG   000006          C.PRI   000010          OP.NLT= ****** GX       $FNPRO= ****** GX       $TMLST= ****** GX


. ABS.  000020    000   (RW,I,GBL,ABS,OVR)
        000270    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8920 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.25
SY:SETPRO.V2,[132,134]SETPRO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETPRO
```

```
                                                                     D  4
SETPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE            REFERENCES

CO.KNC  = ******  GX      5-157       5-220
CS.DTE  = ******  GX      6-280
CS.PVC  = ******  GX      5-140       5-155
C$LENX  = ******  GX      5-136       5-137
C.STS   = ******  GX      5-155      *5-189      *5-199       6-280
D$TADR  = ******  GX      6-284
ERRALL  = ******  GX      5-239
ERRCHN    000145 R       #4-95        5-233
ERRDTE    000045 R       #4-93        5-251
ERRPAR    000106 R       #4-94        5-236
ERRPRM    000210 R       #4-96        7-335
ERRPVC    000000 R       #4-92        5-242
EX$ERR  = 000002         #4-91        4-92        4-93        4-94        4-95        4-96
FL.CRE  = ******  GX      5-123       5-245
FL.DLM  = ******  GX      5-161
FOUND     000324 R       #4-103      *5-122      *5-176       5-224
OC.CHN  = ******  GX      5-128       5-177
OC.CTM  = ******  GX      5-201
OC.DTE  = ******  GX      5-125       5-214
OC.MBL  = ******  GX      5-181
OC.MWI  = ******  GX      5-191
OC.STA  = ******  GX      5-207
PRMERR    001316 R        5-185       5-195      #7-324
PV$PSZ  = ******  GX      5-189
PV$WSZ  = ******  GX      5-199
P$VCT   = ******  GX     *5-205
P$VLCN  = ******  GX     *5-179
P$VLEN  = ******  GX      5-132       5-248
P$VNAM  = ******  GX      5-159       5-163       5-169       5-173       7-328       7-330
P$VOWN  = ******  GX      5-203
P$VPSZ  = ******  GX     *5-188
P$VSTA  = ******  GX     *5-209      *5-212
P$VWSZ  = ******  GX     *5-198
SCR       000326 R       #4-104       7-326
SETDTE    001212 R        5-216      #6-276
SW.ENA  = ******  GX      5-210
TXDAT     000272 R       #4-97        4-98        5-186
TXDATL  = 000014         #4-98        5-186
TXWND     000306 R       #4-99        4-100       5-196
TXWNDL  = 000016         #4-100       5-196
WC.CNT  = ******  GX      5-167       5-222
WC.TRI  = ******  GX      5-171       5-222
WC.UNT  = ******  GX      5-222
X$LENX  = ******  GX      5-136       5-137
$BUFF   = ******  GX      7-334       7-337
$BYTMN  = ******  GX     *5-136
$BYTXT  = ******  GX     *5-137
$CFERR  = ******  GX      5-253
$CHN    = ******  GX      5-179
$CIRID  = ******  GX      5-141       5-142       5-159       5-163
$CNTNM  = ******  GX      5-169
$COMP   = ******  GX      5-157       5-220
```

```
                                  .SBTTL  SETxxx - ASSORTED ACTION ROUTINES
                          ;+
                          ; SETWOR - MOVE A WORD QUANTITY
                          ;
                          ; INPUTS:         R0 = SOURCE ADDRESS
                          ;                 R1 = DESTINATION ADDRESS
                          ;                 R2 = SIZE IN WORDS
                          ;                 R3 = BIT ENCODED FLAGS
                          ;
                          ; REGISTERS:      R0, R1 AND R2 ARE CLOBBERED.
                          ;-

000640  066701  177176    SETWOR: ADD     SERPT,R1                ; POINT TO DESTINATION
000644  012021            10$:    MOV     (R0)+,(R1)+             ; MOVE A WORD
000646  077202                    SOB     R2,10$                  ; ...
000650  016701  177166            MOV     SERPT,R1                ; SET FLAG BIT
000654  050361  000004            BIS     R3,S$FLG(R1)
000660  000207                    RETURN


                          ;+
                          ; SETBIT - SET A BIT ENCODED VALUE
                          ;
                          ; INPUTS:         R0      = ADDRESS OF PARSED BITS
                          ;                 R1      = TARGET OFFSET IN SER$DF TEMPLATE
                          ;                 R2      = BIT VALUE OF INTEREST
                          ;
                          ;-
000662  066701  177154    SETBIT: ADD     SERPT,R1        ; POINT TO TARGET
000666  040211                    BIC     R2,(R1)         ; ASSUME BIT NOT PARSED
000670  011000                    MOV     (R0),R0         ; GET ALL PARSED BITS
000672  005102                    COM     R2              ; CLEAR...
000674  040200                    BIC     R2,R0           ; ...DON'T CARE BITS
000676  050011                    BIS     R0,(R1)         ; SET THAT BIT IF PARSED
000700  000207                    RETURN

                          ;+
                          ; SETFIL - UPDATE OR CREATE A FILE ENTRY
                          ;
                          ; INPUTS:         R0 = ADDRESS OF PROTO FILE BLOCK
                          ;                 R1 = OFFSET IN FIL$DF
                          ;
                          ;
                          ; OUTPUT:         IF NECESSARY A FILE BLOCK IS CREATED.
                          ;                 THE FILE BLOCK IS LINKED INTO THE FIL$DF TEMPLATE.
                          ;
                          ;-
000702                    SETFIL: SAVRG   R5                      ; SAVE TABLE POINTER
000704  066701  177134            ADD     FILPT,R1                ; POINT TO DESTINATION
000710  005711                    TST     (R1)                    ; WAS THERE A PREVIOUS FILE ?
000712  001402                    BEQ     10$                     ; IF EQ, NO
000714  004767  000060            CALL    DEAFIL                  ; ELSE DEALLOCATE FILE

000720  004767  000150    10$:    CALL    FNDFIL                  ; TRY TO FIND PARSED FILE
000724  010511                    MOV     R5,(R1)                 ; POINT FIL$DF TO FILE BLOCK
000726  103020                    BCC     20$                     ; IF CC, GOT IT

000730  004767  000210            CALL    ALCFIL                  ; ELSE ALLOCATE AN EMPTY FILE BLOCK
```

```
                .TITLE  SETROU - SET EXECUTOR ROUTING OPTIONS
                .IDENT  /V05.00/
                .ENABL  LC
;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SET ROUTING PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;


```

```
                                            .SBTTL  $SESLT - SET SLT PARAMETERS
                                    ;+
                                    ; *** - $SESLT - SET SLT PARAMETERS
                                    ;
                                    ; INPUT:
                                    ;       $OPRND = OPERANDS PRESENT
                                    ;       $SWICH = SWITCH WORD
                                    ;       $DVNAM = THE DEVICE NAME ( IF = 0, THEN ALL PROCESSES )
                                    ;       $CNTNM = THE CONTROLLER NUMBER
                                    ;       $UNTNM = THE UNIT NUMBER
                                    ;       $STAN = STATION NUMBER
                                    ;
                                    ; OUTPUT:
                                    ;       SLT TEMPLATE MODIFIED
                                    ;
                                    ;-

000146                              $SESLT::
000146  032767  000000G 000000G             BIT     #FL.PVC,$FLAGS  ; IS CIRCUIT A PVC ?
000154  001076                              BNE     40$             ; IF NE, YES - NOTHING TO DO HERE
000156  012705  000000G                     MOV     #$TMLST,R5      ; GET START OF TEMPLATES
000162  022767  000000G 000000G             CMP     #CO.LIN,$COMP   ; OPERATING ON A LINE ?
000170  001403                              BEQ     5$              ; IF EQ, YES
000172  052767  000000G 000000G             BIS     #FL.TRI,$FLAGS  ; ELSE LOOK FOR TRIB IN .FNSLT/$FNSTA CALLS
000200  004767  001130              5$:     CALL    CHKSTA          ; CHECK FOR ENDNODE ANOMOLIES
000204  103002                              BCC     10$             ; IF CC, OK
000206  000167  000140                      JMP     40$             ; GET OUT WHILE THE GETTIN'S GOOD
                                    ;
                                    ; SEARCH FOR THE RIGHT SLT
                                    ;
000212  004767  000000G             10$:    CALL    .FNSLT          ; GET NEXT SLT
000216  103455                              BCS     40$             ; IF CS, NO MORE

000220  052765  000000G 000000G     20$:    BIS     #CS.MOD,C.STS(R5) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED

000226  032767  000000G 000000G             BIT     #OC.CTM!OL.CTM,$OPRND   ; CHANGE COUNTER TIMER ?
000234  001406                              BEQ     25$                     ; IF EQ, NO
000236  016765  000000G 000000G             MOV     $CT,S$LCTT(R5)  ; UPDATE COUNTER TIMER
000244  052765  000000G 000000G             BIS     #SL$CTM,C.STS(R5) ; MARK COUNTER TIMER PRESENT

000252  032767  000000G 000000G     25$:    BIT     #OC.MXR,$OPRND  ; MAXIMUM ROUTERS PRESENT ?
000260  001403                              BEQ     26$             ; IF EQ, NO
000262  016765  000000G 000000G             MOV     $CIMXR,S$LBRA(R5) ; ELSE UPDATE TEMPLATE

000270  032767  000000G 000000G     26$:    BIT     #OC.RPR,$OPRND  ; ROUTER PRIORITY ?
000276  001403                              BEQ     28$             ; NO
000300  016765  000000G 000000G             MOV     $CIRPR,S$LRPR(R5) ; ELSE UPDATE TEMPLATE
                                    ;
                                    ; LOOK FOR OPTIONS WHICH APPLY TO SLT$DF IF PTP, STA$DF IF MTP
                                    ;
                                    ; (STATE, HELLO TIMER, SERVICE)
                                    ;
000306  032767  000000G 000000G     28$:    BIT     #SW.SEN!SW.SDI,$SWICH ; SERVICE [ENABLED/DISABLED] SPECIFIED ?
000314  001004                              BNE     29$                   ; YES
000316  032767  000000G 000000G             BIT     #OC.STA!OL.STA!OC.HTM,$OPRND ; CHANGE THE STATE/HELLO TIMER ?
000324  001411                              BEQ     35$                   ; IF EQ, NO .. LOOK FOR MORE
000326  022767  000000G 000000G     29$:    CMP     #CO.LIN,$COMP   ; CHANGE OF LINE STATE ?
```

```
                                                                         D  8
SETSLT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24       PAGE 2
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE             REFERENCES

S$LHTM  = ******  GX     *7-318
S$LRPR  = ******  GX     *4-164
S$LTYP  = ******  GX      5-215     7-313
S$TFLG  = ******  GX      11-498
WC.CIR  = ******  GX      11-483
$BUFF   = ******  GX      10-428    10-443
$CFERR  = ******  GX      10-443    11-501
$CIMXR  = ******  GX      4-160
$CIRPR  = ******  GX      4-164
$CNFIG  = ******  GX      11-464
$COMP   = ******  GX      4-139     4-174     7-293
$CT     = ******  GX      4-155
$EDMSG  = ******  GX      10-442
$FLAGS  = ******  GX      4-136    *4-141     8-347     9-396     10-436    11-458    11-477    11-487   *11-509
$FNSTA  = ******  GX      8-352     11-479    11-495
$HTM    = ******  GX      7-318     8-366
$OPRND  = ******  GX      4-153     4-158     4-162     4-172     7-291     7-307     8-355     8-363     11-462
$SAVRG  = ******  GX      8-346     11-457
$SAVVR  = ******  GX      10-427
$SESLT    000146 RG      #4-135
$STAN   = ******  GX      11-459   *11-508
$SWICH  = ******  GX      4-170     5-204     5-213     5-226     6-260     7-297     7-300     7-303     7-323
                          7-326     8-359     8-369     8-372     11-466
$TMLST  = ******  GX      4-138     11-471    11-482
$WILD   = ******  GX      11-460    11-468   *11-483   *11-507
.FNSLT  = ******  GX      4-148     11-474    11-485
```

E 8

Table of contents

```
     5-  145    $SEUNT - SET UNIT PARAMETERS
```

D 10

```
SETX2P - SET X.25 LEVEL 2 PARAM MACRO V05.03b Saturday 29-Jun-85 05:24
Table of contents
```

```
     5-    88    $SEX2P - SET X.25 LEVEL 2 PARAMETERS
```

E 10

```
SETX2P - SET X.25 LEVEL 2 PARAM MACRO V05.03b Saturday 29-Jun-85 05:24  Page 3
```

```
 000264  116760  000000G 000000G          MOVB    $RET,X$3RT(R0)  ; UPDATE VALUE

 000272  032767  000000G 000000G 80$:     BIT     #XP.RST,$OPRND  ; RESTART TIMER?
 000300  001403                           BEQ     90$             ; BR IF NO
 000302  116760  000000G 000000G          MOVB    $RST,X$3ST(R0)  ; UPDATE VALUE

 000310  000207                  90$:     RETURN
                                 ;
                                 ; ERRORS
                                 ;
 000312                          101$:    ERROR$  #ERRX3P         ; NO X3P$DF IN SYSTEM
 000322  000261                           SEC
 000324  000207                           RETURN
         000001                           .END
```

```
SHOBJ      CREATED BY  MACRO  ON 29-JUN-85 AT 05:25      PAGE 2

MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

EPRINT           #4-85
ERROR$           #4-74        5-260
OBJDF$           #4-59        4-61
PRINT            #4-67        5-211        5-247
```

FILEID**SHOCIR

```
                                                .SBTTL  GETSLT - GET THE LOADING INFORMATION CO-ROUTINE
                                        ;+
                                        ; *** - GETSLT - GET THE LOADING INFORMATION CO-ROUTINE
                                        ;
                                        ; THIS CO-ROUTINE IS CALLED UNTIL A C-BIT IS SET. THIS CO-ROUTINE RETURNS
                                        ; TRIBUTARY STATUS INFORMATION FOR EACH TRIBUTRARY.
                                        ;
                                        ; NOTE - THIS CO-ROUTINE USES R5 AS TEMPORARY STORAGE, THEREFORE THE CALLING
                                        ;        ROUTINE MUST NOT DESTROY R5.
                                        ;
                                        ; INPUTS:
                                        ;       $DVNAM = LINE DESCRIPTOR BLOCK
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       STABLK = THE LINE STATE ARGUMENT BLOCK
                                        ;       HELBLK+0 = HELLO TIMER (MTP)
                                        ;       HELBLK+2,+4 = SERVICE (MTP)
                                        ;-

003112                                  GETSLT:
003112  011646                                  MOV     (SP),-(SP)          ; MOVE RETURN ADDRESS UP ON STACK
003114  010566  000002                          MOV     R5,2(SP)            ; SAVE R5
003120  016705  174654                          MOV     CURSLT,R5           ; POINT AT CURRENT SLT
003124  004767  000000G                         CALL    $FNSLT              ; FIND THE NEXT SLT TEMPLATE
003130  103573                                  BCS     70$                 ; IF CS, NOT FOUND
                                        ;
                                        ; FOUND IT !!
                                        ;
003132  010567  174642                  10$:    MOV     R5,CURSLT           ; SAVE CURRENT SLT ADDRESS
003136  016567  000000G 176602                  MOV     S$LHTM(R5),HELBLK   ; SET UP HELLO TIMER
003144  012767  000007  176576                  MOV     #ENABL,HELBLK+2     ; ASSUME SERVICE ENABLED
003152  012767  001420' 176572                  MOV     #ENAB,HELBLK+4      ; ...
003160  032765  000000G 000000G                 BIT     #LF.SER,S$LFLG(R5)  ; SERVICE ENABLED ?
003166  001406                                  BEQ     15$                 ; IF EQ, YES
003170  012767  000010  176552                  MOV     #DSABL,HELBLK+2     ; ELSE SERVICE DISABLED
003176  012767  001427' 176546                  MOV     #DSAB,HELBLK+4      ; ...
003204  005067  176626                  15$:    CLR     STABLK+6            ; ASSUME POINT TO POINT LINE
003210  005067  176626                          CLR     STABLK+12           ; ...
003214  005067  176624                          CLR     STABLK+14           ; ... MUST ZERO THE ADDRESS
003220  012767  000003  176604                  MOV     #OFFL,STABLK+2      ; ELSE, ASSUME IT'S OFF
003226  012767  001330' 176600                  MOV     #OFF,STABLK+4       ; ...
003234  032767  000000G 000000G         20$:    BIT     #FL.MTP,$FLAGS      ; IS THE LINE MULTI-POINT ?
003242  001502                                  BEQ     40$                 ; IF EQ, NO
003244  032767  000000G 000000G                 BIT     #FL.TRI,$FLAGS      ; DO WE NEED TRIBUTARY INFO ?
003252  001510                                  BEQ     60$                 ; IF EQ, NO
003254  004767  000000G                         CALL    $FNSTA              ; FIND THE NEXT STATION TEMPLATE
003260  103517                                  BCS     70$                 ; IF CS, NO MORE STATIONS
                                        ;
                                        ; STATION COST, HELLO TIMER, SERVICE
                                        ;
003262  116567  000000G 176540                  MOVB    S$TCST(R5),STABLK   ; ELSE, GET LINE COST
003270  016567  000000G 176450                  MOV     S$THTM(R5),HELBLK   ; SET UP HELLO TIMER
003276  012767  000007  176444                  MOV     #ENABL,HELBLK+2     ; ASSUME SERVICE ENABLED
003304  012767  001420' 176440                  MOV     #ENAB,HELBLK+4      ; ...
003312  132765  000000G 000000G                 BITB    #SF.SER,S$TFLG(R5)  ; SERVICE ENABLED ?
003320  001406                                  BEQ     25$                 ; IF EQ, YES
```

```
                                                                           D 14
SHOCIR      CREATED BY  MACRO  ON 29-AUG-85 AT 11:51        PAGE 2
SYMBOL CROSS REFERENCE                                      CREF   04.00

SYMBOL  VALUE               REFERENCES

FMT2R5      000402 R        #4-154      12-794
GETSLT      003112 R         6-369      #8-497
GETSTA      004322 R         10-688     #11-726     12-865
GETTCI      002724 R         5-329      #7-436
HEADR       000002 R        #4-144       5-265
HEADR1      000050 R        #4-145       5-268
HELBLK      001746 R         4-217      #4-239      *8-507      *8-508      *8-509      *8-512      *8-513      *8-529      *8-530
                            *8-531      *8-534      *8-535
HELMSG      000224 R        #4-150       4-217       4-233
HWP         001544 R        #4-200       4-210       7-440
INCM        001410 R        #4-190       4-190      10-663
INCML     = 000010          #4-190      10-662
LF.BRO    = ******    GX     6-405
LF.ENA    = ******    GX     8-561
LF.SER    = ******    GX     8-510
LF.X2P    = ******    GX     6-372
LINID       000137 R        #4-147       9-591      12-800      12-802      12-805
LINMSG      000124 R        #4-146       6-400
L$CNT     = ******    GX     9-596
L$NAM     = ******    GX     7-441       9-593
L$TRI     = ******    GX     9-610
L$UNT     = ******    GX     9-602
NF.END    = ******    GX     6-407
NONE        001343 R        #4-185       4-185      12-807
NONEL     = 000006          #4-185      12-806
NSTLIN      001302 R        #4-178
OFF         001330 R        #4-183       4-183       8-518      10-686      12-834
OFFL      = 000003          #4-183       8-517      10-685      12-833
ON          001326 R        #4-182       4-182       8-564      10-693      12-830
ONL       = 000002          #4-182       8-563      10-692      12-829
OPTMSG      000163 R        #4-148
OUTCIR      002602 R         6-379       6-384      #6-398
OUTG        001400 R        #4-189       4-189      10-667
OUTGL     = 000010          #4-189      10-666
PERM        001367 R        #4-188       4-188      12-823
PERML     = 000011          #4-188      12-822
PRINT       005240 R         6-403      10-709      12-878      #13-895
PROTBL      001550 R        #4-203       7-448       7-449
PVCBLK      001754 R         4-221      #4-241      12-774      12-799      12-803
PVCBL2      001760 R         4-222      #4-242
PVCBL3      002002 R         4-223      #4-243
PVCBL4      002012 R         4-224      #4-244
PVCBL5      002020 R         4-225      #4-245
PVCBL6      002022 R         4-226      #4-246      12-868
PVCLIN      000663 R        #4-162      12-848
PVCMSG      000420 R        #4-156       4-221
PVCMS2      000437 R        #4-157       4-222
PVCMS3      000504 R        #4-158       4-223
PVCMS4      000564 R        #4-160       4-224
PVCMS5      000625 R        #4-161       4-225
PVCMS6      000707 R        #4-163       4-226      12-784      12-867
PVC01       000537 R        #4-159      *12-777     *12-783
```

```
                                                                           E 14
SHOCIR      CREATED BY  MACRO  ON 29-AUG-85 AT 11:51        PAGE 3
```

```
002330  012022                                  MOV     (R0)+,(R2)+             ; LOAD UP TYPE STRING ADDRESS
002332  012700  001432'                         MOV     #OFF,R0                 ; ASSUME VERIFICATION IS "TURNED" OFF
002336  005703                                  TST     R3                      ; DID WE FIND THE NSP FEATURES TEMPLATE ?
002340  001411                                  BEQ     160$                    ; IF EQ, NO
002342  032763  000001  000002G                 BIT     #NF.ACC,C.PRM+2(R3) ; IS ACCESS VERIFICATION PRESENT ?
002350  001405                                  BEQ     160$                    ; IF EQ, NO
002352  032763  100000  000006G                 BIT     #NF$ACC,C.PRM+6(R3) ; IS VERIFICATION ON ?
002360  001401                                  BEQ     160$                    ; IF EQ, NO .. OKAY
002362  022020                                  CMP     (R0)+,(R2)+             ; ELSE, CHANGE MESSAGE
002364  012022                          160$:   MOV     (R0)+,(R2)+             ; VERIFICATION STATE LENGTH
002366  012022                                  MOV     (R0)+,(R2)+             ; AND STRING ADDRESS
002370  112767  000054  176166                  MOVB    #<',>,VFYCOM            ; ASSUME ROUTING NODE
002376  016705  177054                          MOV     ROUTMP,R5               ; COPY ROUTING TEMPLATE ADDRESS
002402  016522  000004                          MOV     C.ADD(R5),(R2)+ ; STORE MAXIMUM ADDRESS
002406  001002                                  BNE     170$                    ; IF NE, NOT AN END NODE
002410  105067  176150                          CLRB    VFYCOM                  ; ELSE DON'T PRINT MAX ADDRESS
002414  012701  000516'                 170$:   MOV     #VFYMSG,R1              ; POINT AT INPUT STRING
002420  004767  000320                          CALL    PRINT
002424  004767  000426                          CALL    SEGSIZ                  ; GET ECL SEGMENT SIZE
                                        ;
                                        ; MAX HOPS AND MAX COST
                                        ;
002430  105767  176130                          TSTB    VFYCOM                  ; SEE IF JUST PRINTED ROUTING INFO
002434  001450                                  BEQ     190$                    ; IF EQ, NO - DON'T WANT HOPS AND COST
002436  012702  001460'                         MOV     #MSGBLK,R2              ; POINT TO ARGUMENT BLOCK
002442  016705  177010                          MOV     ROUTMP,R5               ; COPY ROUTING TEMPLATE ADDRESS
002446  016522  000006                          MOV     C.CST(R5),(R2)+ ; STORE MAXIMUM COST
002452  016522  000010                          MOV     C.HOP(R5),(R2)+ ; STORE MAXIMUM HOPS
002456  016522  000012                          MOV     C.RTM(R5),(R2)+ ; ROUTING TIMER
002462  016522  000016                          MOV     C.NBEA(R5),(R2)+ ; BROADCAST ENDNODE ADJACENCIES
002466  016522  000024                          MOV     C.L2TM(R5),(R2)+ ; BROADCAST ROUTING TIMER
002472  012702  001460'                         MOV     #MSGBLK,R2              ; POINT TO ARGUMENT BLOCK
002476  012701  000646'                         MOV     #ROUMSG,R1              ; POINT AT INPUT STRING
002502  004767  000236                          CALL    PRINT
                                        ;
                                        ; LEVEL 2 ROUTERS PARAMETERS
                                        ;
002506  032767  000100  000000G         180$:   BIT     #NF.LV2,$CNFIG  ; CONFIGURED AS LEVEL 2 ROUTER ?
002514  001420                                  BEQ     190$                    ; IF EQ, NOPE
002516  012702  001460'                         MOV     #MSGBLK,R2              ; POINT TO ARGUMENT BLOCK
002522  016705  176730                          MOV     ROUTMP,R5               ; COPY ROUTING TEMPLATE ADDRESS
002526  016522  000014                          MOV     C.NA(R5),(R2)+          ; NUMBER OF AREAS
002532  016522  000020                          MOV     C.AHOP(R5),(R2)+ ; AREA MAX HOPS
002536  016522  000022                          MOV     C.ACST(R5),(R2)+ ; AREA MAX COST
002542  012702  001460'                         MOV     #MSGBLK,R2              ; POINT TO ARGUMENT BLOCK
002546  012701  001076'                         MOV     #RO2MSG,R1              ; POINT AT INPUT STRING
002552  004767  000166                          CALL    PRINT                   ; PRINT IT
                                        ;
                                        ; FOR PSI SYSTEMS, EXECUTOR SUBADDRESS(ES)
                                        ;
002556  016700  000000G                 190$:   MOV     $PSIPT,R0               ; POINT TO PSI STUFF
002562  001436                                  BEQ     220$                    ; IF EQ, NO PSI HERE BOSS
002564  032760  000000C 000000G                 BIT     #CS.TLO!CS.THI,C.STS(R0) ; ANY SUBBADDRESSES ?
002572  001432                                  BEQ     220$                    ; IF EQ, YES WE HAVE NO SUBADDRESSES
002574  012702  001460'                         MOV     #MSGBLK,R2              ; ELSE POINT TO ARG LIST
002600  012701  001201'                         MOV     #SUBBA2,R1              ; ASSUME NO LOW SUBBADDRESS
002604  112711  000040                          MOVB    #<' >,(R1)              ; WANT LEADING BLANK IN HI MSG
```

```
                ;****
                ; LOCAL MACROS
                ;****

                        .MACRO  PRINT   TEXT
                        .IF DIF <TEXT><R0>
                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                        .ENDC
                        CALL    $PRINT          ; PRINT MESSAGE
                        .ENDM   PRINT

                        .MACRO  ERROR$  TEXT
                ;
                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                ; MESSAGE STRING.
                ;
                        .IF DIF <TEXT><R0>
                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                        .ENDC
                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                        .ENDM   ERROR$

                        .MACRO  EPRINT  TEXT
                ;
                ; PRINT TEXT ON ERROR LUN
                ;
                        .IF DIF <TEXT><R0>
                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                        .ENDC
                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                        .ENDM   EPRINT

                ;
                ; DEFINE A PROTOCOL STRING
                ;
                        .MACRO  PRO     A
                        .NCHR   $$$,<A>
                        .PSECT  TEXT,D
                $$$1 = .
                        .ASCII  /A/
                        .PSECT
                        .WORD   $$$,$$$1
                        .ENDM

                ;****
                ; LIBRARY MACRO CALLS
                ;****

                        .MCALL  CHADF$

        000000          CHADF$                  ; DEFINE THE CHARACTERISTICS WORDS OFFSETS

                ;****
                ; LOCAL DATA
                ;****

                ;
```

Table of contents

```
     4-    53    MACRO DEFINITIONS
     5-   124    $SEOBJ - UPDATE OBJECT DEFINITION
     6-   232    FINOBJ - FIND NEXT OBJECT IN SORTED LIST
     7-   256    CHKMUL - CHECK FOR MULTICOPY TASKNAME
```

```
SETPAR - SET PARAMETER  MACRO V05.03b Saturday 29-Jun-85 17:50
Table of contents
```

```
     5-   55    SETPAR - SET A PARAMETER AND DATA TYPE
```

```
                                                              E  3
SETPRO     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23     PAGE 1

SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE            REFERENCES

CL.LEN  = 000020          #4-57
CS.MOD  = ******   GX      5-117
C.CTM     000016          #4-57
C.DEV     000012          #4-57
C.EXT     000015          #4-57
C.FLG     000006          #4-57       *5-147     *5-150
C.LIN     000011          #4-57
C.NAM     000004          #4-57        5-125
C.NCTD  = 000013          #4-57       *5-141
C.NLT     000014          #4-57       *5-131
C.NLTD  = 000012          #4-57       *5-138
C.PRI     000010          #4-57
C.STS   = ******   GX     *5-117
ERRNSP    000000 R        #4-91        5-128
EX$ERR  = 000002          #4-90        4-91
FL.LLC  = ******   GX      5-118
OP.NCT  = ******   GX      5-139
OP.NLT  = ******   GX      5-123       5-136
OP.STA  = ******   GX      5-145
SW.LOA  = ******   GX      5-148
ZF.MFL  = ******   GX      5-147       5-150
$CFERR  = ******   GX      5-128
$FLAGS  = ******   GX      5-118
$FNPRO  = ******   GX      5-112
$NCT    = ******   GX      5-141
$NLT    = ******   GX      5-131       5-138
$OPRND  = ******   GX      5-123       5-136      5-139      5-145
$SEPRO    000076 RG       #5-107
$SWICH  = ******   GX      5-148
$TMLST  = ******   GX      5-108
```

```
                                                              F  3
SETPRO     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23     PAGE 2
```

```
                                                                      E  4
SETPVC     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE              REFERENCES

$CT      = ******   GX      5-205
$DTE     = ******   GX      6-282
$EDMSG   = ******   GX      7-336
$FLAGS   = ******   GX      5-123     5-161     5-245
$MBL     = ******   GX      5-183     5-188     5-193
$MWI     = ******   GX      5-198
$OPRND   = ******   GX      5-125     5-128     5-177     5-181     5-191     5-201     5-207     5-214
$PRINT   = ******   GX      7-337
$PSIPT   = ******   GX      5-151
$RLBF    = ******   GX      5-250
$RQBF    = ******   GX      5-133
$SEPVC     000346 RG       #5-118
$STAN    = ******   GX      5-173
$SWICH   = ******   GX      5-210
$TMLST   = ******   GX      6-277    *6-297    *6-301
$WILD    = ******   GX      5-167     5-171     5-222
$XXAVL   = ******   GX      5-131     5-249
```

```
000734  010511                          MOV     R5,(R1)                 ; POINT FIL$DF TO NEW FILE BLOCK
000736  010501                          MOV     R5,R1                   ; COPY FILE BLOCK POINTER
000740  005721                          TST     (R1)+                   ; SKIP LINK
000742  005021                          CLR     (R1)+                   ; CLEAR USE COUNT
000744  012702  000037                  MOV     #F.LEN-<F.DEV-1>,R2     ; BYTE COUNT
000750  112021                  15$:    MOVB    (R0)+,(R1)+             ; MOVE PROTO TO FILE BLOCK
000752  077202                          SOB     R2,15$                  ; ...
000754  004767  000236                  CALL    GETBYT                  ; GET SIZE OF BYTE ALLOCATION
000760  060067  000000G                 ADD     R0,$BYTMN               ; UPDATE MINIMUM BYTE ALLOCATION
000764  060067  000000G                 ADD     R0,$BYTXT               ; AND ACTUAL BYTE ALLOCATION

000770  005265  000002          20$:    INC     F.CNT(R5)               ; UP USE COUNT
000774                                  RESRG   R5                      ; RESTORE TABLE POINTER
000776  000207                          RETURN

```

```
                                        ;****
                                        ; LOCAL MACROS
                                        ;****

                                                .MACRO  PRINT   TEXT
                                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                                CALL    $PRINT          ; PRINT MESSAGE
                                                .ENDM   PRINT

                                                .MACRO  ERROR$  TEXT
                                        ;
                                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                        ; MESSAGE STRING.
                                        ;
                                                .IF DIF <TEXT><R0>
                                                MOV     TEXT,R0                         ; GET ADDRESS OF ERROR MESSAGE
                                                .ENDC
                                                CALL    $CFERR                          ; PRINT ERROR MESSAGE
                                                .ENDM   ERROR$

                                                .MACRO  EPRINT  TEXT
                                        ;
                                        ; PRINT TEXT ON ERROR LUN
                                        ;
                                                .IF DIF <TEXT><R0>
                                                MOV     TEXT,R0                         ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                                .ENDC
                                                CALL    $CFER1                          ; PRINT MESSAGE ON ERROR LUN
                                                .ENDM   EPRINT

                                        ;****
                                        ; LOCAL DATA
                                        ;****

                                                .NLIST  BEX
                                                .ENABL  LC
        000002                                  EX$ERR = 2
000000     002     103     106          ERRROU: .ASCIZ  <EX$ERR>/CFE -- No routing definition/
000036     002     103     106          ERRPNA: .ASCIZ  <EX$ERR>/CFE -- Routing parameters not applicable/
                                                .EVEN
                                        ;
                                        ; DEFINE ROUTING TEMPLATE
                                        ;
000000                                          .ASECT
        000000                                  .=0
000000                                          .BLKW   1               ; C.LNK
000002                                          .BLKW   1               ; C.STS
000004                                  C.ADD:  .BLKW   1               ; MAXIMUM NODE ADDRESS
000006                                  C.CST:  .BLKW   1               ; MAXIMUM COST ALLOWED
000010                                  C.HOP:  .BLKW   1               ; MAXIMUM NUMBER OF HOPS ALLOWED
000012                                  C.RTM:  .BLKW   1               ; ROUTING TIMER
000014                                  C.NA:   .BLKW   1               ; NUMBER OF AREAS
000016                                  C.NBEA: .BLKW   1               ; NUMBER BRO ENDNODE ADJACENCIES
000020                                  C.AHOP: .BLKW   1               ; AREA HOPS
000022                                  C.ACST: .BLKW   1               ; AREA COST
000024                                  C.L2TM: .BLKW   1               ; LEVEL 2 TIMER
        000026                          C.LEN=.
```

```
    175 000334  001003                          BNE     30$             ; IF NE NO
    176 000336  004767  000012                  CALL    SETLIN          ; SET LINE STATE
    177 000342  000402                          BR      35$             ; CONTINUE
    178 000344  004767  000134          30$:    CALL    SETCIR          ; ELSE SET CIRCUIT STATE
    179 000350  000720                  35$:    BR      10$             ; AND LOOK FOR MORE WORK
    180
    181 000352  000207                  40$:    RETURN
    182
```

```
                                                                          E  8
SETSLT     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24      PAGE 3
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME        REFERENCES

CALLR             #3-56      5-198      6-253
EPRINT            #3-74
ERROR$            #3-63      10-443     11-501
PRINT             #3-58
```

```
                                                                          F  8
**FILE**ID**SETSVC
```

```
                .TITLE  SETUNT - SET LINE UNIT OPTIONS
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - SET UNIT PARAMETERS
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  14-DEC-79
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;
```

```
                                        .TITLE  SETX2P - SET X.25 LEVEL 2 PARAMETERS
                                        .IDENT  /V05.00/
                                        .ENABL  LC

                                  ;
                                  ; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
                                  ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                  ;
                                  ;
                                  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                  ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                  ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                  ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                  ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                  ; TRANSFERRED.
                                  ;
                                  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                  ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                  ; CORPORATION.
                                  ;
                                  ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                  ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                  ;
                                  ;
                                  ; MODULE DESCRIPTION:
                                  ;
                                  ; CFE - SET X.25 LEVEL 2 PARAMETERS
                                  ;
                                  ;
                                  ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                  ;
                                  ; IDENT HISTORY:
                                  ;
                                  ; 1.00  26-OCT-81
                                  ;       DECNET-11M/S V3.1
                                  ;       DECNET-11M-PLUS V1.1
                                  ;
                                  ; 3.00  16-APR-82
                                  ;       DECNET-11M V3.1
                                  ;       DECNET-11M-PLUS V1.1
                                  ;
                                  ; 4.00  07-NOV-83
                                  ;       DECNET-11M V4.0
                                  ;       DECNET-11M-PLUS V2.0
                                  ;
                                  ; 5.00  22-JUL-85
                                  ;       DECnet-11M/S V4.2
                                  ;       DECnet-11M-Plus V3.0
                                  ;       DECnet-Micro/RSX V1.0
                                  ;
                                  ;
```

```
                                                E 11
SETX3P - SET X.25 LEVEL 3 PARAM MACRO V05.03b Saturday 29-Jun-85 05:25  Page 6-2
Symbol table

CS.X3P= ****** GX       XP.MXB= ****** GX       X$3DW = ****** GX       X$3ST = ****** GX       $MWI  = ****** GX
C.STS = ****** GX       XP.MXC= ****** GX       X$3KM = ****** GX       $BSZ  = ****** GX       $MXC  = ****** GX
ERRX3P  000000R         XP.MXR= ****** GX       X$3KT = ****** GX       $CALT = ****** GX       $MXR  = ****** GX
EX$ERR= 000002          XP.MXW= ****** GX       X$3MW = ****** GX       $CFERR= ****** GX       $OPRND= ****** GX
XP.CLT= ****** GX       XP.RET= ****** GX       X$3MXP= ****** GX       $CLT  = ****** GX       $PSIPT= ****** GX
XP.CTM= ****** GX       XP.RST= ****** GX       X$3RM = ****** GX       $DFD  = ****** GX       $RET  = ****** GX
XP.DFB= ****** GX       X$3CT = ****** GX       X$3RT = ****** GX       $DFW  = ****** GX       $RST  = ****** GX
XP.DFW= ****** GX       X$3DFP= ****** GX       X$3SM = ****** GX       $MRE  = ****** GX       $SEX3P  000036RG
XP.MRE= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000326    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8425 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.02
SY:SETX3P.V2,[132,134]SETX3P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETX3P
```

```
                                                F 11
SETX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 05:25      PAGE 1
```

E 12

```
**FILE**ID**SHOCIR

  SSSSSSSS   HH      HH     OOOOOO       CCCCCCCC     IIIIII     RRRRRRRR
  SSSSSSSS   HH      HH     OOOOOO       CCCCCCCC     IIIIII     RRRRRRRR
SS           HH      HH   OO      OO   CC               II       RR      RR
SS           HH      HH   OO      OO   CC               II       RR      RR
SS           HH      HH   OO      OO   CC               II       RR      RR
SS           HH      HH   OO      OO   CC               II       RR      RR
  SSSSSS     HHHHHHHHHH   OO      OO   CC               II       RRRRRRRR
  SSSSSS     HHHHHHHHHH   OO      OO   CC               II       RRRRRRRR
        SS   HH      HH   OO      OO   CC               II       RR  RR
        SS   HH      HH   OO      OO   CC               II       RR  RR
        SS   HH      HH   OO      OO   CC               II       RR    RR          ....
        SS   HH      HH   OO      OO   CC               II       RR    RR          ....
SSSSSSSS     HH      HH     OOOOOO       CCCCCCCC     IIIIII     RR      RR        ....
SSSSSSSS     HH      HH     OOOOOO       CCCCCCCC     IIIIII     RR      RR        ....


LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

F 12

SHOCIR - LIST CIRCUIT CHARACTER MACRO V05.03b Thursday 29-Aug-85 11:51

```
    534 003322  012767  000010  176420          MOV     #DSABL,HELBLK+2 ; ELSE SERVICE DISABLED
    535 003330  012767  001427' 176414          MOV     #DSAB,HELBLK+4  ; ...
    536                                 ;
    537                                 ; TRIBUTARY ADDRESS
    538                                 ;
    539 003336  116501  000001G         25$:    MOVB    C.PRM+1(R5),R1  ; GET TRIBUTARY ADDRESS
    540 003342  042701  177400                  BIC     #^C<377>,R1     ; DEFEAT SIGN EXTENSION
    541 003346  012700  001457'                 MOV     #TRIVAL,R0      ; POINT TO OUTPUT BUFFER FOR TRIBUTARY
    542 003352  012702  014012                  MOV     #10.+<4000*3>,R2; BASE 10. CONVERSION
    543 003356  004767  000000G                 CALL    $CBTA           ; CONVERT TO ACII
    544 003362  162700  001437'                 SUB     #TRIB,R0        ; CALCULATE LENGTH OF STRING
    545 003366  010067  176444                  MOV     R0,STABLK+6     ; STORE LENGTH IN ARGUMENT BLOCK
    546                                 ;
    547                                 ; ACTIVE RATIO
    548                                 ;
    549 003372  032767  000000G 000000G         BIT     #FL.DLC,$FLAGS  ; IS THIS A COMBINATION DDM/DLC ?
    550 003400  001017                          BNE     30$             ; IF NE, YES .. SKIP ACTIVE RATIO
    551 003402  116501  000004G                 MOVB    C.PRM+4(R5),R1  ; GET ACTIVE RATIO
    552 003406  012700  001520'                 MOV     #APRVAL,R0      ; POINT TO OUTPUT BUFFER FOR ACTIVE RATIO
    553 003412  012702  014012                  MOV     #10.+<4000*3>,R2 ; BASE 10. CONVERSION
    554 003416  004767  000000G                 CALL    $CBTA           ; CONVERT TO ACII
    555 003422  162700  001471'                 SUB     #APR,R0         ; CALCULATE LENGTH OF STRING
    556 003426  010067  176410                  MOV     R0,STABLK+12    ; STORE LENGTH IN ARGUMENT BLOCK
    557 003432  012767  001471' 176404          MOV     #APR,STABLK+14  ; AND ADDRESS OF STRING
    558
    559 003440  132765  000000G 000002G 30$:    BITB    #SF.ENA,C.PRM+2(R5) ; IS THE STATION OFF ?
    560 003446  000403                          BR      50$             ; CONTINUE
    561 003450  032765  000000G 000002G 40$:    BIT     #LF.ENA,C.PRM+2(R5) ; IS THE LINE OFF ?
    562 003456  001406                  50$:    BEQ     60$             ; IF EQ, YES .. OKAY
    563 003460  012767  000002  176344          MOV     #ONL,STABLK+2   ; ELSE, IT'S ON
    564 003466  012767  001326' 176340          MOV     #ON,STABLK+4    ; ...
    565 003474  000241                  60$:    CLC                     ; SUCCESS !!
    566 003476  004736                          CALL    @(SP)+          ; CALL THE CALLER BACK
    567
    568 003500  032767  000000G 000000G         BIT     #FL.MTP,$FLAGS  ; IS THE LINE MULTI-POINT ?
    569 003506  001410                          BEQ     80$             ; IF EQ, NO LEAVE
    570 003510  032767  000000G 000000G         BIT     #FL.TRI,$FLAGS  ; WANT ANY MORE TRIBUTARY INFO ?
    571 003516  001205                          BNE     10$             ; IF NE, YES
    572 003520  016605  000002          70$:    MOV     2(SP),R5        ; ELSE, RESTORE R5
    573 003524  012616                          MOV     (SP)+,(SP)      ; MOVE RETURN ADDRESS DOWN ON STACK
    574 003526  000401                          BR      90$             ; AND RETURN
    575 003530  012605                  80$:    MOV     (SP)+,R5        ; ELSE, RESTORE R5
    576 003532  000261                  90$:    SEC                     ; ALL DONE
    577 003534  000207                          RETURN
```

```
                                                                            E 14
SHOCIR    CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 3
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE              REFERENCES

PVCSW       001666 R        #4-226      *12-778     *12-784     *12-861     *12-867
PVCTBL      001642 R        #4-221       12-877
PV$PSZ   =  ******   GX      12-836
PV$WSZ   =  ******   GX      12-841
P$VCT    =  ******   GX      12-825
P$VLCN   =  ******   GX      12-824
P$VNAM   =  ******   GX      12-756      12-760      12-765      12-769      12-772      12-773      12-787      12-788      12-790
                             12-795      12-796
P$VOWN   =  ******   GX      12-775      12-808      12-862
P$VPSZ   =  ******   GX      12-838
P$VSTA   =  ******   GX      12-831
P$VWSZ   =  ******   GX      12-843
RPRMSG      000325 R        #4-153       6-410
SCRACH      001532 R        #4-199
SERMSG      000246 R        #4-151      *6-401      *10-643     *12-786     *12-797
SERV        001355 R        #4-187       4-187
SERVL    =  000012          #4-187
SF$INC   =  000002           10-664
SF.ENA   =  ******   GX      8-559       10-690
SF.SER   =  ******   GX      8-532
SHOPVC      004360 R         5-352      #12-753
SHOSVC      003672 R         5-354      #10-630
STABLK      002030 R         4-218      #4-248       6-411      *7-471      *8-514      *8-515      *8-516      *8-517      *8-518
                            *8-528      *8-545      *8-556      *8-557      *8-563      *8-564
STAMSG      000266 R        #4-152       4-218
SVCBLK      001726 R         4-229       4-230       4-231       4-232       4-233       4-234      #4-237       10-640
SVCM1       000745 R        #4-165       4-229
SVCM2       000773 R        #4-167       4-230
SVCM3       001050 R        #4-168       4-231
SVCM4       001121 R        #4-169       4-232
SVCM5       001173 R        #4-170       4-234
SVCO1       001223 R        #4-171      *10-697     *10-700
SVCTBL      001674 R        #4-229       10-708
SV$DTE   =  ******   GX      10-698
SV$PSZ   =  ******   GX      10-669
SV$WSZ   =  ******   GX      10-674
S$LBRA   =  ******   GX      6-413
S$LFLG   =  ******   GX      6-372       6-405       8-510
S$LHTM   =  ******   GX      8-507
S$LRPR   =  ******   GX      6-412
S$TCST   =  ******   GX      8-528       10-694      12-869
S$TFLG   =  ******   GX      8-532       10-690
S$THTM   =  ******   GX      8-529       10-695      12-871
S$VFLG   =  ******   GX      10-664
S$VNAM   =  ******   GX      10-641      10-642      10-650      10-654      10-660      10-661
S$VPSZ   =  ******   GX      10-671
S$VRCL   =  ******   GX      10-678
S$VWSZ   =  ******   GX      10-676
TRIB        001437 R        #4-195       4-249       8-544
TRIVAL      001457 R        #4-196       8-541
TRNTBL      001624 R        #4-216       6-402
TYPBLK      001736 R         4-216      #4-238      *7-448      *7-449      *7-463      *7-464      *7-466      *7-467
```

```
                                                                            F 14
SHOCIR    CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 4
```

```
    415 002610  032760  000000G 000000G         BIT     #CS.TLO,C.STS(R0) ; DO I HEAR A LOW SUBADDRESS ?
    416 002616  001414                          BEQ     200$              ; IF EQ, NO - GO GET HIGH AND PRINT
    417 002620  016022  000000G                 MOV     P$STLO(R0),(R2)+  ; ELSE SAVE VALUE,
    418 002624  105011                          CLRB    (R1)              ; ASSUME NO HIGH SUBADDRESS MISSING
    419 002626  012701  001154'                 MOV     #SUBBA,R1         ; USE CORRECT FORMATTING STRING
    420 002632  032760  000000G 000000G         BIT     #CS.THI,C.STS(R0) ; IS THERE A HIGH SUBADDRESS ?
    421 002640  001405                          BEQ     210$              ; IF EQ, NOPE
    422 002642  112767  000054  176331          MOVB    #<',>,SUBBA2      ; USE CORRECT SEPARATOR
    423 002650  016012  000000G         200$:   MOV     P$STHI(R0),(R2)   ; ELSE SAVE IT IN LIST
    424 002654  004767  000064          210$:   CALL    PRINT             ; PRINT IT
    425
    426 002660  000207                  220$:   RETURN
    427                                 ;
    428                                 ; ERROR CONDITIONS
    429                                 ;
    430 002662                          230$:   ERROR$  #ERRNOD           ; FAILED TO FIND NODE TEMPLATE
    431 002672  000423                          BR      280$              ; AND RETURN
    432 002674                          240$:   ERROR$  #ERRXPT           ; FAILED TO FIND NSP TEMPLATE
    433 002704  000416                          BR      280$              ;
    434 002706                          250$:   ERROR$  #ERRROU           ; NO ROUTING TEMPLATE
    435 002716  000411                          BR      280$
    436 002720                          260$:   ERROR$  #ERRECL           ; NO ECL TEMPLATE
    437 002730  000404                          BR      280$
    438 002732                          270$:   ERROR$  #ERRDEC           ; NO DEC$DF TEMPLATE
    439 002742  000207                  280$:   RETURN
```

```
                        ; LOCAL EQUATES
                        ;
       000001                   OP.CNT  = 1             ; DISPLAY CONTROLLER INFO.
       000002                   OP.UNT  = 2             ; DISPLAY UNIT INFO.
       000004                   OP.OPT  = 4             ; DISPLAY OPTIONAL INFO
       000010                   OP.STA  = 10            ; DISPLAY LINE STATE INFO
       000020                   OP.X2P  = 20            ; DISPLAY X.25 LEVEL 2 INFO

                        ;
                        ; DEFINE THE ASYNCHRONOUS BAUD RATE TABLE
                        ;
000000                  RATETB:                         ; BIT SETTINGS
000000 000062                   .WORD   50.             ; 0
000002 000113                   .WORD   75.             ; 1
000004 000156                   .WORD   110.            ; 2
000006 000206                   .WORD   134.            ; 3
000010 000226                   .WORD   150.            ; 4
000012 000454                   .WORD   300.            ; 5
000014 001130                   .WORD   600.            ; 6
000016 002260                   .WORD   1200.           ; 7
000020 003410                   .WORD   1800.           ; 10
000022 003720                   .WORD   2000.           ; 11
000024 004540                   .WORD   2400.           ; 12
000026 007020                   .WORD   3600.           ; 13
000030 011300                   .WORD   4800.           ; 14
000032 016040                   .WORD   7200.           ; 15
000034 022600           END:    .WORD   9600.           ; 16
                                                        ; 17 - > 9600 BAUD
       000034                   TBLEN=<END-RATETB>      ; CALCULATE LENGTH OF TABLE

                        ;
                        ; UNIBUS RUN MASK TABLE
                        ;
000036                  URMTBL:                         ; BIT
000036 000761'                  .WORD   CPA             ; 0
000040 000764'                  .WORD   CPB             ; 1
000042 000767'                  .WORD   CPC             ; 2
000044 000772'                  .WORD   CPD             ; 3
000046 000775'                  .WORD   UBE             ; 4
000050 001000'                  .WORD   UBF             ; 5
000052 001003'                  .WORD   UBH             ; 6
000054 001006'                  .WORD   UBJ             ; 7
000056 001011'                  .WORD   UBK             ; 10
000060 001014'                  .WORD   UBL             ; 11
000062 001017'                  .WORD   UBM             ; 12
000064 001022'                  .WORD   UBN             ; 13
000066 001025'                  .WORD   UBP             ; 14
000070 001030'                  .WORD   UBR             ; 15
000072 001033'                  .WORD   UBS             ; 16
000074 001036'                  .WORD   UBT             ; 17

000076                  CURSLT: .BLKW   1               ; CURRENT SLT TEMPLATE ADDRESS
000100                  DSPOPT: .BLKW   1               ; DISPLAY OPTIONS
                        ;
                        ; DEFINE UNIT TEMPLATE
                        ;
000000                          .ASECT
```

```
                        .TITLE  SETOBJ - SET OBJECT OPTIONS
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - MODULE TO REPLACE MODIFIED FILEDS IN THE OBJECT TEMPLATE
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;

```

```
        .TITLE  SETPAR - SET PARAMETER
        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       NETWORK MANAGEMENT - SET PARAMETER
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  08-JUL-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 2.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

F 3

```
SETPRO     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

EPRINT           #4-78
ERROR$           #4-67        5-128
LLCTP$           #4-55        4-57
PRINT            #4-62
```

G 3

```
**FILE**ID**SETIPVC
```

```
                                                                           F  4
SETPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 3

MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME       REFERENCES

CALLR            #4-83
EPRINT           #4-73
ERROR$           #4-62        5-253
PRINT            #4-57        7-337
RESRG            #4-83        7-338
SAVRG            #4-83        7-325       7-333
```

```
                                                                           G  4
**FILE**ID**SETREM
```

```
                                        .SBTTL  DEAFIL-DEALLOCATE A FILE NAME BLOCK

                                ;+
                                ; DEAFIL-DEALLOCATE A FILE NAME BLOCK
                                ;
                                ; INPUTS:       (R1) = ADDRESS OF FILE NAME BLOCK
                                ;
                                ; OUTPUT:       TARGET NAME BLOCK USE COUNT IS DECREMENTED.
                                ;               IF NECESSARY IT IS DEALLOCATED AND $BYTMN,$BYTXT
                                ;               ARE ADJUSTED.
                                ;
                                ; ALL REGISTERS ARE PRESERVED.
                                ;-
001000  004767  000000G         DEAFIL: CALL    $SAVAL                  ; SAVE EM
001004  011105                          MOV     (R1),R5                 ; GET NAME BLOCK ADDRESS
001006  005365  000002                  DEC     F.CNT(R5)               ; REDUCE USE COUNT
001012  101025                          BHI     30$                     ; IF HI, CAN'T DEALLOCATE
001014  012704  000000G                 MOV     #$FILHD,R4              ; ELSE POINT TO LISTHEAD
001020  010403                  10$:    MOV     R4,R3                   ; SAVE PREDECESSOR
001022  011404                          MOV     (R4),R4                 ; POINT TO NEXT NAME BLOCK
001024  001421                          BEQ     101$                    ; IF EQ, SOMETHING'S WRONG
001026  020405                          CMP     R4,R5                   ; IS THIS THE TARGET BLOCK ?
001030  001373                          BNE     10$                     ; IF NE, KEEP LOOKING
001032  011413                          MOV     (R4),(R3)               ; REMOVE IT FROM THE LIST
001034  012700  000000G                 MOV     #$XXAVL,R0              ; POINT TO FREE LIST
001040  012701  000043                  MOV     #F.LEN,R1               ; SIZE OF BLOCK
001044  010502                          MOV     R5,R2                   ; POINT TO BLOCK
001046  004767  000000G                 CALL    $RLBF                   ; RELEASE CORE BLOCK
001052  004767  000140                  CALL    GETBYT                  ; GET SIZE OF ALLOCATION IN BYTES
001056  160067  000000G                 SUB     R0,$BYTMN               ; REDUCE BYTE POOL MINIMUM
001062  160067  000000G                 SUB     R0,$BYTXT               ; ..AND PRESENT VALUE
001066  000207                  30$:    RETURN

001070  000167  000000G         101$:   JMP     .CONER                  ; BIG TROUBLE

```

```
    111 000110                                .PSECT
    112
```

```
                                                .SBTTL  SETLIN - SET LINE STATE
                                        ;+
                                        ; ** SETLIN - SET LINE STATE
                                        ;
                                        ; INPUTS        R5 = ADDRESS OF SLT TEMPLATE
                                        ;
                                        ; OUTPUT        SLT TEMPLATE MODIFIED
                                        ;
                                        ; REGISTERS     NO REGISTERS ARE MODIFIED
                                        ;-

 000354                                 SETLIN:
 000354  032765  000000G 000000G                BIT     #LF.MTP,S$LFLG(R5) ; MULTIPOINT LINE ?
 000362  001002                                 BNE     10$                ; BR IF YES
 000364                                         CALLR   SETPTP             ; ELSE SET POINT TO POINT STATE
                                        ;
                                        ; MULTIPOINT LINE: TREAT REQUEST FOR ON AS REQUEST FOR OFF
                                        ;
 000370  032765  000000G 000000G        10$:    BIT     #LF.MFL,S$LFLG(R5) ; IS THIS LINE LOADED ?
 000376  001010                                 BNE     20$                ; IF NE, YES
 000400  032767  000000G 000000G                BIT     #SW.LOA,$SWICH     ; NEW STATE ON OR OFF ?
 000406  001435                                 BEQ     40$                ; IF EQ, NO - ALREADY CLEARED
 000410  052765  000000G 000000G                BIS     #LF.MFL,S$LFLG(R5) ; ELSE NEW STATE = OFF
 000416  000431                                 BR      40$
                                        ;
                                        ; M-P CONTROLLER: DON'T ALLOW CHANGE FROM ON TO [OFF,CLEARED]
                                        ;                     IF ANY CIRCUITS ARE STILL ACTIVE
                                        ; M-P TRIBUTARY:  TURN CORRESPONDING CIRCUIT OFF
                                        ;
 000420  032767  000000G 000000G        20$:    BIT     #SW.ENA,$SWICH     ; NEW STATE = ON ?
 000426  001025                                 BNE     40$                ; IF NE, YES - LEAVE CURRENT STATE (ON OR OFF)
 000430  105765  000000G                        TSTB    S$LTYP(R5)         ; CHECK FOR MULTIPOINT CONTROLLER
 000434  001002                                 BNE     22$                ; IF NE, YES - DON'T CHANGE CIRCUITS
 000436  004767  000340                         CALL    SETSTA             ; ELSE TURN CORRESPONDING CIRCUIT OFF
 000442  004767  000512                 22$:    CALL    FNDACT             ; ELSE CHECK FOR TRIBUTARY ACTIVITY
 000446  103003                                 BCC     30$                ; IF CC, NO CIRCUITS ACTIVE
 000450  004767  000552                 25$:    CALL    ACTERR             ; ELSE CAN'T CHANGE STATE WITH CIRCUITS ACTIVE
 000454  000412                                 BR      40$                ; CONTINUE
                                        ;
                                        ; MULTIPOINT LINE: NO TRIBS ARE ACTIVE, SET NEW STATE OF OFF OR CLEARED.
                                        ;
 000456  042765  000000G 000000G        30$:    BIC     #LF.ENA!LF.MFL,S$LFLG(R5) ; ASSUME NEW STATE = CLEARED
 000464  032767  000000G 000000G                BIT     #SW.LOA,$SWICH     ; MARK FOR LOAD ?
 000472  001403                                 BEQ     40$                ; IF EQ, NO
 000474  052765  000000G 000000G                BIS     #LF.MFL,S$LFLG(R5) ; ELSE MARK IT FOR LOAD

 000502  000207                         40$:    RETURN
```

```
**FILE**ID**SETSVC




  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  VV      VV    CCCCCCCC
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  VV      VV    CCCCCCCC
SS          EE              TT      SS          VV      VV  CC
SS          EE              TT      SS          VV      VV  CC
SS          EE              TT      SS          VV      VV  CC
SS          EE              TT      SS          VV      VV  CC
  SSSSSS    EEEEEEEE        TT        SSSSSS    VV      VV  CC
  SSSSSS    EEEEEEEE        TT        SSSSSS    VV      VV  CC
        SS  EE              TT              SS  VV      VV  CC
        SS  EE              TT              SS  VV      VV  CC
        SS  EE              TT              SS    VV  VV    CC             ....
        SS  EE              TT              SS    VV  VV    CC             ....
SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS        VV        CCCCCCCC     ....
SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS        VV        CCCCCCCC     ....



LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                ;****
                                ; LIBRARY MACRO CALLS
                                ;****

                                        .MCALL  CHADF$

000000                                  CHADF$                  ; DEFINE CHARACTERISTICS WORDS OFFSETS

                                ;****
                                ; LOCAL DATA
                                ;****

000000                          DDM:    .BLKW   1               ; DDM PROCESS NAME
                                        .NLIST  BEX
                                        .ENABL  LC
        000002                          EX$ERR = 2
        000000                          EX$WAR = 0
000002     002     103     106  ERRBAU: .ASCIZ  <EX$ERR>/CFE -- Illegal baud rate/
000034     000     103     106  ERRSEC: .ASCIZ  <EX$WAR>/CFE -- No unit CSR on this line/
                                        .EVEN
                                ;
                                ; DEFINE THE ASYNCHRONOUS BAUD RATE TABLE
                                ;
000076                          RATETB:                         ; BIT SETTINGS
000076  000062                          .WORD   50.             ; 0
000100  000113                          .WORD   75.             ; 1
000102  000156                          .WORD   110.            ; 2
```

```
                        ;****
                        ; LOCAL MACROS
                        ;****

                                .MACRO  PRINT   TEXT
                                MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                CALL    $PRINT          ; PRINT MESSAGE
                                .ENDM   PRINT

                                .MACRO  ERROR$  TEXT
                        ;
                        ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                        ; MESSAGE STRING.
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                .ENDC
                                CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                .ENDM   ERROR$

                                .MACRO  EPRINT  TEXT
                        ;
                        ; PRINT TEXT ON ERROR LUN
                        ;
                                .IF DIF <TEXT><R0>
                                MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                .ENDC
                                CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                .ENDM   EPRINT
```

```
                                                                  F 11
SETX3P      CREATED BY  MACRO  ON 29-JUN-85 AT 05:25       PAGE 1
SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL  VALUE                REFERENCES

CS.X3P  = ******  GX      6-113
C.STS   = ******  GX      6-113
ERRX3P    000000 R       #5-92          6-163
EX$ERR  = 000002         #5-90          5-92
XP.CLT  = ******  GX      6-119
XP.CTM  = ******  GX      6-115
XP.DFB  = ******  GX      6-123
XP.DFW  = ******  GX      6-127
XP.MRE  = ******  GX      6-143
XP.MXB  = ******  GX      6-135
XP.MXC  = ******  GX      6-139
XP.MXR  = ******  GX      6-147
XP.MXW  = ******  GX      6-131
XP.RET  = ******  GX      6-151
XP.RST  = ******  GX      6-155
X$3CT   = ******  GX     *6-117
X$3DFP  = ******  GX     *6-125
X$3DW   = ******  GX     *6-129
X$3KM   = ******  GX     *6-141
X$3KT   = ******  GX     *6-121
X$3MW   = ******  GX     *6-133
X$3MXP  = ******  GX     *6-137
X$3RM   = ******  GX     *6-145
X$3RT   = ******  GX     *6-153
X$3SM   = ******  GX     *6-149
X$3ST   = ******  GX     *6-157
$BSZ    = ******  GX      6-137
$CALT   = ******  GX      6-117
$CFERR  = ******  GX      6-163
$CLT    = ******  GX      6-121
$DFD    = ******  GX      6-125
$DFW    = ******  GX      6-129
$MRE    = ******  GX      6-145
$MWI    = ******  GX      6-133
$MXC    = ******  GX      6-141
$MXR    = ******  GX      6-149
$OPRND  = ******  GX      6-115     6-119     6-123     6-127     6-131     6-135     6-139     6-143     6-147
                          6-151     6-155
$PSIPT  = ******  GX      6-110
$RET    = ******  GX      6-153
$RST    = ******  GX      6-157
$SEX3P    000036 RG      #6-109
```

```
                                                                  G 11
SETX3P      CREATED BY  MACRO  ON 29-JUN-85 AT 05:25       PAGE 2
```

Table of contents

```
     5-  252     $SHCIR - SHOW CIRCUIT CHARACTERISTICS
     6-  357     DISPLY - DISPLAY TRANSPORT CIRCUIT INFO
     7-  419     GETTCI - GET TRANSPORT CIRCUIT INFORMATION
     8-  477     GETSLT - GET THE LOADING INFORMATION CO-ROUTINE
     9-  579     FMTLIN - FORMAT A LINE ID
    10-  616     SHOSVC - SHOW SWITCHED VIRTUAL CIRCUITS
    11-  716     GETSTA - GET STATION TEMPLATE
    12-  737     SHOPVC - SHOW PERMANENT VIRTUAL CIRCUIT
    13-  884     PRINT - PRINT FORMATTED DATA
```

```
                                               .SBTTL  FMTLIN - FORMAT A LINE ID
                                       ;+
                                       ; *** - FMTLIN - FORMAT A LINE ID
                                       ;
                                       ; INPUTS:
                                       ;       $FLAGS = FLAGS WORD
                                       ;
                                       ; OUTPUTS:
                                       ;       LINE ID FORMATTED INTO OUTPUT BUFFER
                                       ;
                                       ;-

003536  012700  000137'                FMTLIN: MOV     #LINID,R0               ; GET ADDRESS OF OUTPUT BUFFER
003542  012703  000000G                FMTLN1: MOV     #$DVNAM,R3              ; POINT AT LINE DESCRIPTOR
003546  016301  000000G                FMTLN2: MOV     L$NAM(R3),R1            ; GET DEVICE NAME
003552  004767  000000G                        CALL    $C5TA                   ; CONVERT NAME TO ASCII
003556  112720  000055                         MOVB    #'-,(R0)+               ; INSERT DELIMETER
003562  116301  000000G                        MOVB    L$CNT(R3),R1            ; GET CONTROLLER NUMBER
003566  012702  014012                         MOV     #<10.!<3*4000>>,R2      ; BASE=10., WIDTH = 3
003572  004767  000000G                        CALL    $CBTA                   ; CONVERT TO ASCII
003576  032767  000000G 000000G                BIT     #FL.MUX,$FLAGS          ; IS THIS A MUX DEVICE ?
003604  001410                                 BEQ     10$                     ; IF EQ, NO
003606  112720  000055                         MOVB    #'-,(R0)+               ; INSERT DELIMETER
003612  116301  000000G                        MOVB    L$UNT(R3),R1            ; ELSE, GET UNIT NUMBER
003616  012702  014012                         MOV     #<10.!<3*4000>>,R2      ; BASE=10., WIDTH = 3
003622  004767  000000G                        CALL    $CBTA                   ; CONVERT TO ASCII
003626  032767  000000G 000000G        10$:    BIT     #FL.MTP,$FLAGS          ; IS THE LINE MULTI-POINT ?
003634  001414                                 BEQ     20$                     ; IF EQ, NO
003636  032767  000000G 000000G                BIT     #FL.SLV,$FLAGS          ; MULTIPOINT SLAVE CIRCUIT ?
003644  001010                                 BNE     20$                     ; IF NE, YES - NO TRIB NUMBER WANTED
003646  112720  000056                         MOVB    #'.,(R0)+               ; INSERT DELIMETER
003652  016301  000000G                        MOV     L$TRI(R3),R1            ; ELSE, GET TRIBUTARY NUMBER
003656  012702  014012                         MOV     #<10.!<3*4000>>,R2      ; BASE=10., WIDTH = 3
003662  004767  000000G                        CALL    $CBTA                   ; CONVERT TO ASCII
003666  105010                         20$:    CLRB    (R0)                    ; MAKE STRING ASCIZ
003670  000207                                 RETURN
```

```
                                                          F 14
SHOCIR     CREATED BY  MACRO  ON 29-AUG-85 AT 11:51     PAGE 4
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE           REFERENCES

TYPMSG    000172 R       #4-149     4-216
WC.CNT  = ******  GX      5-301     5-304     5-317    10-648    12-763    12-781
WC.NAM  = ******  GX      5-266
WC.TRI  = ******  GX     10-652    11-728    12-767    12-781
WC.UNT  = ******  GX      5-324    11-728
X$3DFP  = ******  GX     10-668    12-835
X$3DW   = ******  GX     10-672    12-839
X25       001351 R       #4-186     4-186    12-813
X25L    = 000004         #4-186    12-812
$BUFF   = ******  GX      5-269     5-272     6-409     6-415    13-896    13-901
$CBTA   = ******  GX      8-543     8-554     9-598     9-604     9-612
$CFERR  = ******  GX      6-393
$CIRID  = ******  GX     12-756    12-760
$CNFIG  = ******  GX      6-407
$CNTNM  = ******  GX      5-306    *5-308    10-650   *10-660    12-765   *12-772
$COMP   = ******  GX      5-332    10-644    12-754
$C5TA   = ******  GX      7-442     9-594
$DVNAM  = ******  GX     *5-292     7-441     9-592   *10-659   *12-771
$EDMSG  = ******  GX      5-271     6-414    12-801    13-900
$FLAGS  = ******  GX      5-273     5-283    *5-286     5-322     6-374    *6-383    *6-387    *7-452    *7-459
                          8-519     8-521     8-549     8-568     8-570     9-599     9-605     9-607    10-646
                         12-758    12-779    12-851   *12-852   *12-855   *12-857

$FNPRO  = ******  GX      5-279
$FNSLT  = ******  GX      8-501    11-730
$FNSTA  = ******  GX      8-523    11-732
$GTBUF  = ******  GX      5-270
$PRINT  = ******  GX      5-272     6-400     6-415    13-901
$PSIPT  = ******  GX      5-337
$SHCIR    002046 RG      #5-264
$STAN   = ******  GX     *5-285    *7-460    10-654   *10-661    12-769   *12-773
$TMLST  = ******  GX      5-278     5-293    10-634    11-729
$UNTNM  = ******  GX      5-326    *5-328
$WILD   = ******  GX      5-266     5-301     5-304     5-317      5-32    10-648    10-652    11-727   *11-728
                        *11-733    12-763    12-767    12-781
$$$     = 000017         #4-203     4-203    #4-205     4-205    #4-206     4-206    #4-207     4-207    #4-208
                          4-208    #4-209     4-209    #4-212     4-212    #4-213     4-213    #4-214     4-214
$$$1    = 000065 R       #4-203     4-203    #4-205     4-205    #4-206     4-206    #4-207     4-207    #4-208
                          4-208    #4-209     4-209    #4-212     4-212    #4-213     4-213    #4-214     4-214
```

```
                                                          G 14
SHOCIR     CREATED BY  MACRO  ON 29-AUG-85 AT 11:51     PAGE 5
```

```
                                          .SBTTL  PRINT - PRINT A LINE
                                  ;+
                                  ; **-PRINT-PRINT A LINE
                                  ;
                                  ; INPUTS:         R1 = TEXT FORMATTING STRING
                                  ;                 MSGBLK = ARGUMENT BLOCK
                                  ;
                                  ; OUTPUT:         THE TEXT IS FORMATTED AND PRINTED.
                                  ;
                                  ; REGISTERS:      R0, R1, R2 ARE CLOBBERED.
                                  ;
                                  ;-
002744                            PRINT:
002744  012700  000000G                   MOV     #$BUFF,R0       ; POINT AT OUTPUT BUFFER
002750  012702  001460'                   MOV     #MSGBLK,R2      ; POINT AT THE MESSAGE BLOCK
002754  004767  000000G                   CALL    $EDMSG          ; FORMAT THE LINE
002760                                    PRINT   #$BUFF          ; PRINT MESSAGE
002770  000207                            RETURN
```

```
169             000000                          .=0
170 000000                                      .BLKW   1               ; C.LNK
171 000002                                      .BLKW   1               ; C.STS
172 000004                              C.UNT:  .BLKW   1               ; UNIT NUMBER
173 000006                              C.CWO:  .BLKW   1               ; CHARACTERISTICS WORD 0
174 000010                              C.CW1:  .BLKW   1               ; CHARACTERISTICS WORD 1
175 000012                              C.CSR:  .BLKW   1               ; SECONDARY CSR ADDRESS
176 000014                              C.CST:  .BLKW   1               ; LINE COST
177 000016                              C.DPR:  .BLKW   1               ; DEAD POLLING RATIO
178             000020                  C.LEN=.
179 000102                                      .PSECT
180
181                                             .NLIST  BEX
182                                             .ENABL  LC
183             000002                          EX$ERR = 2
184                                     ;
185                                     ; FORMAT STRINGS - ** NOTE ** EDIT MESSAGE ALWAY MOVES AT LEAST ONE CHARACTER
186                                     ;                  FROM THE INPUT STRING INTO THE OUTPUT BUFFER (EVEN IF YOU
187                                     ;                  SAY THE INPUT LENGTH IS ZERO FOR A %VA) !!
188                                     ;
189 000102      012     114     151     HEADR:  .ASCIZ  <12>/Line characteristics as of %Y %3Z/
190 000145      012     113     156     HEADR1: .ASCIZ  <12>/Known line characteristics as of %Y %3Z/
191 000216      012     114     151     LINMSG: .ASCII  <12>/Line = /
192 000226                              LINID:  .BLKB   20.
193 000252      040     040     103     CNTMSG: .ASCIZ  /  Controller CSR = %P, Vector = %O, Priority = %O%VA/
194 000337      040     040     104     UNTMSG: .ASCIZ  /  Duplex = %VA, Speed = %VA, Protocol = %VA/
195 000413      045     126     101     OPTMSG: .ASCIZ  /%VA%VA/
196 000422      040     040     110     X2PMSG: .ASCIZ  /  Holdback timer = %M, Counter timer = %M/
197 000474      040     040     115     X2PMS2: .ASCIZ  /  Maximum data = %M, Maximum window = %M/
198 000545      040     040     115     X2PMS3: .ASCIZ  /  Maximum retransmits = %M, Retransmit timer = %M00/
199 000631      040     040     123     STAMSG: .ASCIZ  /  State = %VA/
200
201 000647      002     012     103     ERRX2P: .ASCII  <EX$ERR><12>/CFE -- Missing X2P$DF for line /
202 000710                              X2PLIN: .BLKB   20
203                                     ;
204                                     ; REPLACEMENT STRINGS
205                                     ;
206 000730      106     165     154     FULL:   .ASCII  /Full/
207             000004                          FULLN=.-FULL
208 000734      110     141     154     HALF:   .ASCII  /Half/
209             000004                          HALFLN=.-HALF
210 000740      117     156             ON:     .ASCII  /On/
211             000002                          ONL=.-ON
212 000742      117     146     146     OFF:    .ASCII  /Off/
213             000003                          OFFL=.-OFF
214 000745      103     154     145     CLEAR:  .ASCII  /Cleared/
215             000007                          CLEARL=.-CLEAR
216 000754      076     071     066     HISPD:  .ASCII  />9600/
217             000005                          HISPDL=.-HISPD
218
219
220 000761      103     120     101     CPA:    .ASCII  /CPA/
221 000764      103     120     102     CPB:    .ASCII  /CPB/
222 000767      103     120     103     CPC:    .ASCII  /CPC/
223 000772      103     120     104     CPD:    .ASCII  /CPD/
224 000775      125     102     105     UBE:    .ASCII  /UBE/
225 001000      125     102     106     UBF:    .ASCII  /UBF/
```

```
     53                                                          .SBTTL  MACRO DEFINITIONS
     54
     55
     56                                                  ;****
     57                                                  ; LOCAL MACROS
     58                                                  ;****
     59
     60                                                          .MACRO  PRINT   TEXT
     61                                                          MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
     62                                                          CALL    $PRINT          ; PRINT MESSAGE
     63                                                          .ENDM   PRINT
     64
     65                                                          .MACRO  ERROR$  TEXT
     66                                                  ;
     67                                                  ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
     68                                                  ; MESSAGE STRING.
     69                                                  ;
     70                                                          .IF DIF <TEXT><R0>
     71                                                          MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
     72                                                          .ENDC
     73                                                          CALL    $CFERR                  ; PRINT ERROR MESSAGE
     74                                                          .ENDM   ERROR$
     75
     76                                                          .MACRO  EPRINT  TEXT
     77                                                  ;
     78                                                  ; PRINT TEXT ON ERROR LUN
     79                                                  ;
     80                                                          .IF DIF <TEXT><R0>
     81                                                          MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
     82                                                          .ENDC
     83                                                          CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
     84                                                          .ENDM   EPRINT
     85
     86                                                  ;****
     87                                                  ; MACRO CALLS
     88                                                  ;****
     89
     90                                                          .MCALL  ISTAT$,STATE$,TRAN$,OBJDF$
     91
     92 000000                                                  OBJDF$                          ; DEFINE OBJECT BLOCK SYMBOLS
     93
     94                                                  ;****
     95                                                  ; LOCAL DATA
     96                                                  ;****
     97
     98                                                          .NLIST  BEX
     99                                                          .ENABL  LC
    100                 000002                                  EX$ERR = 2
    101 000000             002          103     106     ERRBUF: .ASCIZ  <EX$ERR>/CFE -- Buffer allocation failure/
    102 000042             002          103     106     ERRNAM: .ASCIZ  <EX$ERR>/CFE -- Object task name must be specified/
    103 000115             002          103     106     ERRONA: .ASCIZ  <EX$ERR>/CFE -- Task name not allowed with object type zero/
    104 000201             002          103     106     ERRMUL: .ASCIZ  <EX$ERR>/CFE -- Invalid multicopy taskname/
    105                                                         .EVEN
    106                                                  ;
    107                                                  ; DEFINE OBJECT TEMPLATE
    108                                                  ;
    109 000000                                                  .ASECT
```

```
                                        .SBTTL  SETPAR - SET A PARAMETER AND DATA TYPE
                                ;+
                                ; **-SETPAR-SET A PARAMETER NUMBER AND DATA TYPE
                                ;
                                ; THIS ROUTINE IS CALLED TO SET A PARAMETER NUMBER AND NICE DATA
                                ; TYPE INTO THE OUTPUT BUFFER.
                                ;
                                ; INPUTS:
                                ;       R3 = ADDRESS OF NEXT BYTE IN OUTPUT BUFFER
                                ;       R5 = ADDRESS OF PARAMETER FOLLOWED BY DATA TYPE BYTE
                                ;
                                ; OUTPUTS:
                                ;       R3 = ADDRESS OF NEXT BYTE IN OUTPUT BUFFER
                                ;
                                ; REGISTERS:
                                ;       R4 IS PRESERVED
                                ;-

000000                          SETPAR::
000000  112523                          MOVB    (R5)+,(R3)+             ; SET PARAMETER LOW BYTE
000002  112523                          MOVB    (R5)+,(R3)+             ; AND HIGH BYTE
000004  112523                          MOVB    (R5)+,(R3)+             ; SET DATA TYPE BYTE
000006  105725                          TSTB    (R5)+                   ; SKIP HIGH BYTE
000010  000205                          RTS     R5                      ; RETURN

        000001                          .END
```

```
**FILE**ID**SETPVC




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP     VV      VV     CCCCCCCC
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP     VV      VV     CCCCCCCC
SS           EE               TT       PP      PP   VV      VV   CC
SS           EE               TT       PP      PP   VV      VV   CC
SS           EE               TT       PP      PP   VV      VV   CC
SS           EE               TT       PP      PP   VV      VV   CC
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     VV      VV   CC
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     VV      VV   CC
        SS   EE               TT       PP           VV      VV   CC
        SS   EE               TT       PP           VV      VV   CC
        SS   EE               TT       PP             VV  VV     CC               ....
        SS   EE               TT       PP             VV  VV     CC               ....
SSSSSSSS     EEEEEEEEEE       TT       PP               VV         CCCCCCCC       ....
SSSSSSSS     EEEEEEEEEE       TT       PP               VV         CCCCCCCC       ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                            G 4
**FILE**ID**SETREM





  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   RRRRRRRR     EEEEEEEEEE   MM      MM
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   RRRRRRRR     EEEEEEEEEE   MM      MM
SS           EE               TT       RR      RR   EE           MMMM  MMMM
SS           EE               TT       RR      RR   EE           MMMM  MMMM
SS           EE               TT       RR      RR   EE           MM  MM  MM
SS           EE               TT       RR      RR   EE           MM  MM  MM
  SSSSSS     EEEEEEEE         TT       RRRRRRRR     EEEEEEEE     MM      MM
  SSSSSS     EEEEEEEE         TT       RRRRRRRR     EEEEEEEE     MM      MM
        SS   EE               TT       RR  RR       EE           MM      MM
        SS   EE               TT       RR  RR       EE           MM      MM
        SS   EE               TT       RR    RR     EE           MM      MM       ....
        SS   EE               TT       RR    RR     EE           MM      MM       ....
SSSSSSSS     EEEEEEEEEE       TT       RR      RR   EEEEEEEEEE   MM      MM       ....
SSSSSSSS     EEEEEEEEEE       TT       RR      RR   EEEEEEEEEE   MM      MM       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                            H 4
SETREM    SET REMOTE NODE OPTION MACRO V05.03b Monday 01-Jul-85 16:37
```

```
                                       .SBTTL  FNDFIL - FIND A MATCHING FILE BLOCK

                               ;+
                               ; FNDFIL - FIND A MATCHING FILE BLOCK
                               ;
                               ; INPUTS:         R0      = ADDRESS OF PROTOTYPE FILE BLOCK
                               ;
                               ; OUTPUTS:        IF CS, NO MATCHING FILE BLOCK
                               ;                 OTHERWISE
                               ;
                               ;                 R5      = ADDRESS OF FILE BLOCK IN LIST
                               ;
                               ; ALL REGISTERS ARE PRESERVED.
                               ;-

001074  004767  000000G        FNDFIL: CALL    $SAVAL          ; SAVE EM
001100  012705  000000G                MOV     #$FILHD,R5      ; POINT TO FILE HEADER
001104  011505                 10$:    MOV     (R5),R5         ; NEXT
001106  001414                         BEQ     30$             ; IF EQ, NO MORE
001110  010504                         MOV     R5,R4           ; COPY ADDRESS
001112  062704  000004                 ADD     #F.DEV-1,R4     ; POINT TO TEXT PORTION
001116  010003                         MOV     R0,R3           ; POINT TO PROTO
001120  012702  000037                 MOV     #F.LEN-<F.DEV-1>,R2 ; BYTE COUNT
001124  122324                 20$:    CMPB    (R3)+,(R4)+     ; DO THEY MATCH ?
001126  001366                         BNE     10$             ; IF NE, NO - KEEP LOOKING
001130  077203                         SOB     R2,20$          ; CONTINUE
001132  010566  000014                 MOV     R5,14(SP)       ; UPDATE SAVED R5
001136  005727                         TST     (PC)+           ; SUCCESS
001140  000261                 30$:    SEC                     ; FAILURE
001142  000207                         RETURN
```

```
    114                                                 .SBTTL  $SEROU - SET ROUTING PARAMETERS
    115                                         ;+
    116                                         ; *** - $SEROU - SET ROUTING PARAMETERS
    117                                         ;
    118                                         ; INPUT:
    119                                         ;       $OPRND = OPERANDS PRESENT
    120                                         ;
    121                                         ; OUTPUT:
    122                                         ;       CARRY CLEAR: THE ROUTING TEMPLATE IS MODIFIED.
    123                                         ;       CARRY SET: THE CETAB IS FOR A NON-ROUTING NODE, AND AN ERROR
    124                                         ;                  MESSAGE HAS BEEN PRINTED.
    125                                         ;-
    126
    127 000110                                  $SEROU::
    128 000110  012700  000000G                         MOV     #$TMLST,R0      ; GET START OF TEMPLATES
    129                                         ;
    130                                         ; SEARCH FOR THE ROUTING TEMPLATE
    131                                         ;
    132 000114  011000                          10$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
    133 000116  001522                                  BEQ     101$            ; IF EQ, NO ROUTING TEMPLATE
    134 000120  122760  000000G 000000G                 CMPB    #CS.ROU,C.STS(R0) ; IS THIS THE ROUTING TEMPLATE ?
    135 000126  001372                                  BNE     10$             ; IF NE, NO - KEEP LOOKING
    136 000130  005760  000004                          TST     C.ADD(R0)       ; IS THIS A ROUTING NODE ?
    137 000134  001521                                  BEQ     102$            ; IF EQ, NO - ILLEGAL PARAMETERS
    138 000136  052760  000000G 000000G                 BIS     #CS.MOD,C.STS(R0) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
    139
    140 000144  032767  000000G 000000G                 BIT     #OE.MXH,$OPRND  ; WAS THE MAXIMUM HOPS SPECIFIED ?
    141 000152  001403                                  BEQ     20$             ; IF EQ, NO
    142 000154  016760  000000G 000010                  MOV     $MAXH,C.HOP(R0) ; ELSE, STORE THE NEW HOP MAXIMUM
    143
    144 000162  032767  000000G 000000G 20$:    BIT     #OE.MXC,$OPRND  ; WAS THE MAXIMUM COST SPECIFIED ?
    145 000170  001403                                  BEQ     30$             ; IF EQ, NO
    146 000172  016760  000000G 000006                  MOV     $MAXC,C.CST(R0) ; ELSE, STORE THE NEW COST MAXIMUM
    147
    148 000200  032767  000000G 000000G 30$:    BIT     #OE.NN,$OPRND   ; IS THERE A MAXIMUM NODE ADDRESS ?
    149 000206  001412                                  BEQ     40$             ; IF EQ, NO
    150 000210  026060  000004  000016                  CMP     C.ADD(R0),C.NBEA(R0) ; MAX ADDR = BROAD. ENDNODE ADJ.
    151 000216  001003                                  BNE     35$             ; IF NE, NO
    152 000220  016760  000000G 000016                  MOV     $NN,C.NBEA(R0)  ; ELSE TRACK CHANGES
    153 000226  016760  000000G 000004  35$:    MOV     $NN,C.ADD(R0)   ; STORE THE NEW MAXIMUM NODE ADDRESS
    154
    155 000234  032767  000000G 000000G 40$:    BIT     #OE.RTM,$OPRND  ; WAS THE ROUTING TIMER SPECIFIED ?
    156 000242  001403                                  BEQ     50$             ; IF EQ, NOPE
    157 000244  016760  000000G 000012                  MOV     $RTM,C.RTM(R0)  ; ELSE STORE THE NEW ROUTING TIMER
    158
    159 000252  032767  000000G 000000G 50$:    BIT     #OE.ACOS,$OPRND ; AREA MAX COST ?
    160 000260  001403                                  BEQ     60$             ; NO
    161 000262  016760  000000G 000022                  MOV     $EACOS,C.ACST(R0) ; ELSE UPDATE IT
    162
    163 000270  032767  000000G 000000G 60$:    BIT     #OE.AHOP,$OPRND ; AREA MAX HOPS ?
    164 000276  001403                                  BEQ     70$             ; NOPE
    165 000300  016760  000000G 000020                  MOV     $EAHOP,C.AHOP(R0) ; SAVE IT
    166
    167 000306  032767  000000G 000000G 70$:    BIT     #OE.AMAX,$OPRND ; MAX AREAS
    168 000314  001403                                  BEQ     80$             ; NO
    169 000316  016760  000000G 000014                  MOV     $EAMAX,C.NA(R0) ; SAVE IT
    170
```

```
                                        .SBTTL  SETCIR  - SET CIRCUIT STATE
                                ;+
                                ; ** SETCIR - SET CIRCUIT STATE
                                ;
                                ; INPUTS:       R5 = ADDRESS OF SLT TEMPLATE
                                ;               $OPRND  = PARSED OPTIONS
                                ;               $SWICH  = ADDITIONAL PARSED OPTIONS (SW.SLT BITS)
                                ;
                                ; OUTPUT:       SLT AND STATION TEMPLATES ARE MODIFIED
                                ;
                                ; REGISTERS     NO REGISTERS ARE MODIFIED
                                ;
                                ;-

000504                          SETCIR:
                                ;
                                ; POINT TO POINT CIRCUIT - TREAT SAME AS LINE STATE
                                ;
000504  032765  000000G 000000G         BIT     #LF.MTP,S$LFLG(R5) ; MULTIPOINT CIRCUIT ?
000512  001002                          BNE     10$                ; IF NE, YES
000514                                  CALLR   SETPTP             ; ELSE SET REQUESTED STATE IN SLT
                                ;
                                ; MULTIPOINT - TREAT REQUEST FOR CLEARED SAME AS OFF
                                ;
000520  004767  000256          10$:    CALL    SETSTA             ; SET STATION PARAMETERS
000524  032765  000000G 000000G         BIT     #LF.MFL,S$LFLG(R5) ; IS LINE CURRENTLY CLEARED ?
000532  001004                          BNE     15$                ; IF NE, NO
000534  032767  000000G 000000G         BIT     #SW.ENA,$SWICH     ; ELSE, TURNING CIRCUITS ON ?
000542  001411                          BEQ     20$                ; IF EQ, NO - LEAVE LINE CLEARED
000544  042765  000000G 000000G 15$:    BIC     #LF.ENA,S$LFLG(R5) ; ASSUME ALL CIRCUITS ARE OFF
000552  004767  000402                  CALL    FNDACT             ; ANY ACTIVE CIRCUITS ?
000556  103003                          BCC     20$                ; IF CC, NO - SLT NOT ENABLED
000560  052765  000000G 000000G         BIS     #LF.ENA!LF.MFL,S$LFLG(R5) ; ELSE MARK SLT FOR ENABLE

000566  000207                  20$:    RETURN

```

Table of contents

```
    5-  101    $SESVC - SET SVC PARAMETERS
```

```
000104  000206                   .WORD   134.            ; 3
000106  000226                   .WORD   150.            ; 4
000110  000454                   .WORD   300.            ; 5
000112  001130                   .WORD   600.            ; 6
000114  002260                   .WORD   1200.           ; 7
000116  003410                   .WORD   1800.           ; 10
000120  003720                   .WORD   2000.           ; 11
000122  004540                   .WORD   2400.           ; 12
000124  007020                   .WORD   3600.           ; 13
000126  011300                   .WORD   4800.           ; 14
000130  016040                   .WORD   7200.           ; 15
000132  022600          END:     .WORD   9600.           ; 16
                                                         ; 17  >9600 BAUD
        000034                   TBLEN=<END-RATETB>      ; CALCULATE LENGTH OF TABLE

                        ;
                        ; DEFINE UNIT TEMPLATE
                        ;
000000                           .ASECT
        000000                   .=0
000000                           .BLKW   1               ; C.LNK
000002                           .BLKW   1               ; C.STS
000004                  C.UNT:   .BLKW   1               ; UNIT NUMBER
000006                  C.CW0:   .BLKW   1               ; CHARACTERISTICS WORD 0
000010                  C.CW1:   .BLKW   1               ; CHARACTERISTICS WORD 1
000012                  C.CSR:   .BLKW   1               ; SECONDARY CSR ADDRESS
000014                  C.CST:   .BLKW   1               ; LINE COST
000016                  C.DPR:   .BLKW   1               ; DEAD POLLING RATIO
000020                           .BLKW   1               ; UNUSED
000022                           .BLKW   1               ; UNUSED
        000024          C.LEN=.
000134                           .PSECT
```

```
                                                         .SBTTL  $SEX2P - SET X.25 LEVEL 2 PARAMETERS
                                                ;+
                                                ; *** - $SEX2P - SET X.25 LEVEL 2 PARAMETERS
                                                ;
                                                ; INPUT:
                                                ;        $OPRND = OPERANDS PRESENT
                                                ;
                                                ; OUTPUT:
                                                ;        C-BIT = SUCCESS/FAILURE
                                                ;        THE X2P$DF TEMPLATE IS UPDATED
                                                ;
                                                ;-

000000                                          $SEX2P::
000000  012700  000000G                                  MOV     #$TMLST,R0          ; GET START OF TEMPLATE LIST
                                                ;
                                                ; SEARCH FOR THE LINE
                                                ;
000004  011000                                  10$:     MOV     (R0),R0             ; NEXT TEMPLATE
000006  001500                                           BEQ     80$                 ; IF EQ, NO MORE
000010  122760  000000G 000000G                          CMPB    #CS.X2P,C.STS(R0)   ; IS THIS A X2P$DF TEMPLATE ?
000016  001372                                           BNE     10$                 ; IF NE, NO - KEEP LOOKING
000020  022767  000000G 000000G                          CMP     #WC.LIN,$WILD       ; WILDCARD LINES ?
000026  001424                                           BEQ     20$                 ; BR IF YES
000030  026760  000000G 000000G                          CMP     $DVNAM,X$2LIN(R0)   ; COMPARE WITH PARSED LINE ID
000036  001362                                           BNE     10$                 ; IF NE, WRONG LINE
000040  032767  000000G 000000G                          BIT     #WC.CNT,$WILD       ; WILDCARD CONTROLLERS ?
000046  001004                                           BNE     12$                 ; BR IF YES
000050  026760  000002G 000002G                          CMP     $DVNAM+2,X$2LIN+2(R0) ; COMPARE WITH PARSED LINE ID
000056  001352                                           BNE     10$                 ; IF NE, WRONG LINE
000060  032767  000000G 000000G                 12$:     BIT     #WC.UNT,$WILD       ; WILDCARD UNITS?
000066  001004                                           BNE     20$                 ; BR IF YES
000070  026760  000004G 000004G                          CMP     $DVNAM+4,X$2LIN+4(R0) ; COMPARE WITH PARSED LINE ID
000076  001342                                           BNE     10$                 ; IF NE, WRONG LINE
000100  032767  000000G 000000G                 20$:     BIT     #OL.HBT,$OPRND      ; HOLDBACK TIMER ?
000106  001403                                           BEQ     30$                 ; IF EQ, NO
000110  016760  000000G 000000G                          MOV     $HBT,X$2HBT(R0)     ; ELSE UPDATE
000116  032767  000000G 000000G                 30$:     BIT     #OL.RTC,$OPRND      ; MAXIMUM RETRANSMITS SPECIFIED ?
000124  001403                                           BEQ     40$                 ; IF EQ, NO
000126  016760  000000G 000000G                          MOV     $RTC,X$2RTC(R0)     ; UPDATE
000134  032767  000000G 000000G                 40$:     BIT     #OL.WSZ,$OPRND      ; MAXIMUM WINDOW SPECIFIED?
000142  001403                                           BEQ     50$                 ; IF EQ, NO
000144  016760  000000G 000000G                          MOV     $WSZ,X$2WSZ(R0)     ; UPDATE
000152  032767  000000G 000000G                 50$:     BIT     #OL.BSZ,$OPRND      ; MAXIMUM BLOCK SPECIFIED?
000160  001403                                           BEQ     60$                 ; IF EQ, NO
000162  016760  000000G 000000G                          MOV     $BSZ,X$2BSZ(R0)     ; UPDATE
000170  032767  000000G 000000G                 60$:     BIT     #OL.RTT,$OPRND      ; UPDATE RETRANSMIT TIMER ?
000176  001403                                           BEQ     70$                 ; IF EQ, NOPE
000200  016760  000000G 000000G                          MOV     $RTM,X$2RTT(R0)     ; ELSE UPDATE IT
000206  000676                                  70$:     BR      10$                 ; NEXT LINE

000210  000207                                  80$:     RETURN

        000001                                           .END
```

```
SETX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 05:25       PAGE 2

MACRO CROSS REFERENCE                                      CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-76
ERROR$            #4-65         6-163
PRINT             #4-60
```

```
**FILE**ID**SHOBJ
```

```
		.TITLE	SHOCIR - LIST CIRCUIT CHARACTERISTICS
		.IDENT	/V05.00/
		.ENABL	LC
;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SHOW CIRCUIT CHARACTERISTICS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	15-OCT-81
;	DECNET-11M/S V3.1
;	DECNET-11M-PLUS V1.1
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;

```

```
                                                .SBTTL  SHOSVC - SHOW SWITCHED VIRTUAL CIRCUITS
                                        ;+
                                        ; ** SHOSVC - SHOW SWITCHED VIRTUAL CIRCUIT
                                        ;
                                        ; INPUTS
                                        ;       R3 = ADDRESS OF X3P$DF TEMPLATE
                                        ;
                                        ; OUTPUT
                                        ;       SWITCHED VIRTUAL CIRCUIT CHARACTERISTICS DISPLAYED.
                                        ;
                                        ; REGISTERS
                                        ;       R0, R1, R2 AND R4 ARE CORRUPTED.
                                        ;
                                        ;-

003672                                  SHOSVC:

                                        ; LOCATE THE TEMPLATE OR TEMPLATES OF INTEREST
                                        ;
003672  012705  000000G                         MOV     #$TMLST,R5      ; POINT TO TEMPLATE LISTHEAD
003676  011505                          10$:    MOV     (R5),R5         ; POINT TO NEXT TEMPLATE
003700  001002                                  BNE     11$             ; IF NE, NOT END OF LIST
003702  000167  000412                          JMP     90$             ; BR IF END OF LIST
003706  122765  000000G 000000G         11$:    CMPB    #CS.SVC,C.STS(R5) ; SVC TEMPLATE?
003714  001370                                  BNE     10$             ; BR IF NO
003716  012704  001726'                         MOV     #SVCBLK,R4      ; POINT TO ARG BLOCK
003722  016524  000000G                         MOV     S$VNAM(R5),(R4)+ ; STUFF NAME INTO ARG BLOCK
003726  016524  000002G                         MOV     S$VNAM+2(R5),(R4)+ ; ...
003732  105067  174310                          CLRB    SERMSG          ; SUPPRESS SERVICE PARAMETER
003736  022767  000000G 000000G                 CMP     #CO.KNC,$COMP   ; KNOWN CIRCUITS?
003744  001424                                  BEQ     20$             ; BR IF YES
003746  032767  000000G 000000G                 BIT     #FL.DLM,$FLAGS  ; WAS THIS A VALID DLM FORMAT
003754  001561                                  BEQ     90$             ; NO - MAYBE HE WANTS ''DLM'' PVC
003756  032767  000000G 000000G                 BIT     #WC.CNT,$WILD   ; WILDCARD CONTROLLER
003764  001004                                  BNE     15$             ; BR IF YES
003766  026567  000000G 000000G                 CMP     S$VNAM(R5),$CNTNM ; COMPARE CONTROLLERS
003774  001340                                  BNE     10$             ; BR IF WRONG CONTROLLER
003776  032767  000000G 000000G         15$:    BIT     #WC.TRI,$WILD   ; WILDCARD TRIBUTARIES?
004004  001004                                  BNE     20$             ; BR IF YES
004006  026567  000002G 000000G                 CMP     S$VNAM+2(R5),$STAN ; COMPARE WITH PARSED VALUE
004014  001330                                  BNE     10$             ; BR IF NO MATCH
                                        ;
                                        ; GET SVC PARAMETERS
                                        ;
004016  012767  015355  000000G         20$:    MOV     #^RDLM,$DVNAM   ; SET UP FOR STATION SEARCH
004024  016567  000000G 000000G                 MOV     S$VNAM(R5),$CNTNM ; DITTO
004032  016567  000002G 000000G                 MOV     S$VNAM+2(R5),$STAN ; LIKEWISE
004040  012724  000010                          MOV     #INCML,(R4)+    ; ASSUME INCOMING SVC
004044  012724  001410'                         MOV     #INCM,(R4)+     ; ...
004050  032765  000002  000000G                 BIT     #SF$INC,S$VFLG(R5) ; CHECK SVC TYPE
004056  001006                                  BNE     30$             ; BR IF INCOMING
004060  012764  000010  177774                  MOV     #OUTGL,-4(R4)   ; ELSE TYPE=OUTGOING
004066  012764  001400' 177776                  MOV     #OUTG,-2(R4)    ; ...
004074  016324  000000G                 30$:    MOV     X$3DFP(R3),(R4)+ ; ASSUME MAX DATA OMITTED
004100  032765  000000G 000000G                 BIT     #SV$PSZ,C.STS(R5) ; IS IT THERE ?
004106  001403                                  BEQ     40$             ; IF EQ, NO
004110  016564  000000G 177776                  MOV     S$VPSZ(R5),-2(R4) ; ELSE USE TEMPLATE VALUE
004116  116324  000000G                 40$:    MOVB    X$3DW(R3),(R4)+ ; ASSUME DEFAULT VALUE
```

```
SHOCIR    CREATED BY  MACRO  ON 29-AUG-85 AT 11:51      PAGE 5
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME     REFERENCES

CALLR          #4-108     6-379
CHADF$         #4-108     4-110
EPRINT         #4-75
ERROR$         #4-64      6-393
PRINT          #4-57      5-272     6-400     6-415     13-901
PRO            #4-94      4-203     4-205     4-206     4-207     4-208     4-209     4-212     4-213     4-214
STRING         #4-87      4-182     4-183     4-184     4-185     4-186     4-187     4-188     4-189     4-190
                 4-192    4-193
SVCDF$         #4-108     4-111
```

```
**FILE**ID**SHOEXE
```

```
                                        .SBTTL  PUTADR - PUT A PHASE 4 ADDRESS
                                ;+
                                ; ** PUTADR - PUT A PHASE 4 NODE ADDRESS
                                ;
                                ; CALLING SEQUENCE:     JSR     R5,PUTADDR
                                ;                       .WORD   ADDR1,ADDR2
                                ;
                                ; INPUTS:                 R1 = NODE ADDRESS
                                ;                       (R5) = PHASE 3 FORMAT STRING
                                ;                      2(R5) = AREA FORMAT STRING
                                ;
                                ; OUTPUT:                 R0 = NEW BUFFER POINTER
                                ;                         R1, R2 ARE CORRUPTED.
                                ;-

 002772                         PUTADR:
 002772  010146                         MOV     R1,-(SP)                ; SAVE ADDRESS
 002774  012702  001460'                MOV     #MSGBLK,R2              ; POINT TO ARG LIST
 003000  042701  001777                 BIC     #1777,R1                ; ISOLATE AREA
 003004  042716  176000                 BIC     #^C1777,(SP)            ; ISOLATE NODE ADDRESS
 003010  005701                         TST     R1                      ; IS IT IN AREA 0 ?
 003012  001004                         BNE     10$                     ; IF NE, DISPLAY A LA PHASE 4
 003014  012612                         MOV     (SP)+,(R2)              ; ELSE STUFF IT INTO ARG LIST
 003016  012501                         MOV     (R5)+,R1                ; PICK UP PHASE 3 FORMAT STRING
 003020  005725                         TST     (R5)+                   ; SKIP SECOND ARG
 003022  000412                         BR      20$                     ; ALL DONE

 003024  006201                 10$:    ASR     R1                      ; SHIFT AREA
 003026  006201                         ASR     R1                      ;  INTO HI BYTE
 003030  000301                         SWAB    R1                      ; MAKE THAT LO BYTE
 003032  042701  177700                 BIC     #^C77,R1                ; CLEAR JUNK BITS
 003036  010112                         MOV     R1,(R2)                 ; STUFF AREA INTO ARG LIST
 003040  012662  000002                 MOV     (SP)+,2(R2)             ; STUFF IT INTO LIST
 003044  005725                         TST     (R5)+                   ; SKIP PHASE 3 FORMAT
 003046  012501                         MOV     (R5)+,R1                ; USE NEW IMPROVED PHASE 4 FLAVOR

 003050  004767  000000G        20$:    CALL    $EDMSG                  ; FORMAT IT
 003054  000205                         RTS     R5
```

```
001003    125      102      110  UBH:     .ASCII  /UBH/
001006    125      102      112  UBJ:     .ASCII  /UBJ/
001011    125      102      113  UBK:     .ASCII  /UBK/
001014    125      102      114  UBL:     .ASCII  /UBL/
001017    125      102      115  UBM:     .ASCII  /UBM/
001022    125      102      116  UBN:     .ASCII  /UBN/
001025    125      102      120  UBP:     .ASCII  /UBP/
001030    125      102      122  UBR:     .ASCII  /UBR/
001033    125      102      123  UBS:     .ASCII  /UBS/
001036    125      102      124  UBT:     .ASCII  /UBT/
001041    015      012      040  URM:     .ASCII  <15><12>/  Unibus run mask = [/
001070                           URMVAL:  .BLKB   60.
001164    040      040      115  DPR:     .ASCII  /  Multi-point Dead = /
001211                           DPRVAL:  .BLKB   10.
001223    040      040      125  CSR:     .ASCII  /  Unit CSR = /
001240                           CSRVAL:  .BLKB   10.
001252                           SCRACH:  .BLKB   10.                     ; SCRATCH BUFFER
001264                           HWP:     .BLKB   3                       ; DDM NAME IF HARDWARE PROTOCOL
                                          .EVEN


001270                           PROTBL:  PRO     <None>                  ; 0 NONE
001274  000000  000000                    .WORD   0,0                     ; 1 DDCMP
001300                                    PRO     <Bisync>                ; 2 BYSYNCH
001304                                    PRO     <SDLC>                  ; 3 SDLC
001310                                    PRO     <LAPB>                  ; 4 X.25
001314                                    PRO     <Ethernet>              ; 5 ETHERNET
001320                                    PRO     <????>                  ; 6 Unassigned
001324  000003  001264'                   .WORD   3,HWP                   ; 7 HARDWARE PROTOCOL

001330                           DDCTBL:  PRO     <DDCMP Point>
001334                                    PRO     <DDCMP Tributary>
001340                                    PRO     <DDCMP Control>


001344  000252'                  TBL1:    .WORD   CNTMSG                  ; PRINT THE CONTROLLER INFO
001346  000337'                           .WORD   UNTMSG                  ; PRINT THE UNIT INFO.
001350  000413'                           .WORD   OPTMSG                  ; PRINT THE OPTIONAL INFO.
001352  000422'                           .WORD   X2PMSG                  ; PRINT THE X.25 LEVEL 2 INFO
001354  000474'                           .WORD   X2PMS2                  ;   ...
001356  000545'                           .WORD   X2PMS3                  ;   ...
001360  000631'                           .WORD   STAMSG                  ; PRINT THE STATE INFO.
001362  000000                            .WORD   0

001364  001420' 000001           TBL2:    .WORD   CNTBLK,OP.CNT           ; THE ARGUMENT BLOCK FOR CONTROLLER INFO.
001370  001432' 000002                    .WORD   UNTBLK,OP.UNT           ; THE ARGUMENT BLOCK FOR UNIT INFO.
001374  001446' 000004                    .WORD   OPTBLK,OP.OPT           ; THE OPTIONAL ARGUMENT BLOCK
001400  001466' 000020                    .WORD   X2PBLK,OP.X2P           ; THE X.25 LEVEL 2 ARG BLOCK
001404  001500' 000020                    .WORD   X2PBL2,OP.X2P           ;   ...
001410  001504' 000020                    .WORD   X2PBL3,OP.X2P           ;   ...
001414  001456' 000010                    .WORD   STABLK,OP.STA           ; THE ARGUMENT BLOCK FOR STATE INFO.

001420                           CNTBLK:  .BLKW   5                       ; CONTROLLER ARGUMENT BLOCK
001432                           UNTBLK:  .BLKW   6                       ; UNIT ARGUMENT BLOCK
001446                           OPTBLK:  .BLKW   4                       ; OPTIONAL BLOCK
001456                           STABLK:  .BLKW   4                       ; LOADING ARGUMENT BLOCK
001466                           X2PBLK:  .BLKW   5                       ; X.25 LEVEL 2 BLOCK (LAPB LINES ONLY)
001500                           X2PBL2:  .BLKW   2                       ;   ...
```

```
    110          000000                          .=0
    111 000000                                   .BLKW   1               ; C.LNK
    112 000002                                   .BLKW   1               ; C.STS
    113 000004                           C.TYP:  .BLKB   1               ; OBJECT TYPE
    114 000005                           C.FLG:  .BLKB   1               ; OBJECT FLAG
    115 000006                           C.NAM:  .BLKW   2               ; OBJECT TASK NAME (RAD50)
    116 000012                           C.CPY:  .BLKB   1               ; MAXIMUM NUMBER OF COPIES OF A TASK TO SPAWN
    117                                                                  ; LIMIT IS 63.
    118          000013                  C.LEN=.
    119 000244                                   .PSECT
    120
    121 000244                           OBJNAM: .BLKB   6
    122
```

```
                                                                H  2
SETPAR - SET PARAMETER  MACRO V05.03b Saturday 29-Jun-85 17:50  Page 5-1
Symbol table

A$$CHK= 000000          E$$XPR= 000000          LD$LP = 000000          N$$ACC= 000001          R$$DER= 000000
A$$CPS= 000000          F$$LVL= 000001          L$$ASG= 000000          N$$BUF= 000001          R$$K11= 000001
A$$PRI= 000000          G$$TPP= 000000          L$$DRV= 000000          N$$LDV= 000001          R$$SND= 000000
A$$TRP= 000000          G$$TSS= 000000          L$$P11= 000001          N$$MCP= 000001          R$$11M= 000000
C$$CKP= 000000          G$$TTK= 000000          L$$11R= 000000          N$$MLL= 000001          SETPAR  000000RG
C$$ORE= 000400          G$$WRD= 000000          M$$CRB= 000124          N$$MOV= 000010          S$$WRG= 000000
C$$RSH= 177564          I$$RAR= 000000          M$$CRX= 000000          N$$NCT= 000001          S$$YSZ= 007600
D$$BUG= 177514          I$$RDN= 000000          M$$FCS= 000000          N$$PEM= 000001          T$$KMG= 000000
D$$ISK= 000000          K$$CNT= 177546          M$$MGE= 000000          P$$P45= 000000          T$$MIN= 000000
D$$L11= 000001          K$$CSR= 177546          M$$NET= 000000          P$$WRD= 000000          V$$CTR= 001000
D$$YNC= 000000          K$$LDC= 000000          M$$OVR= 000000          Q$$OPT= 000010          X$$DBT= 000000
D$$YNM= 000000          K$$TPS= 000074

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000012    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8904 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.06
SY:SETPAR.V2,[135,134]SETPAR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[135,10]SETPAR
```

```
                                                         I  2
SETPAR      CREATED BY  MACRO  ON 29-JUN-85 AT 17:50      PAGE 1
```

Table of contents

```
     5-  106    $SEPVC - SET PVC PARAMETERS
     6-  259    SETDTE - SET NEW DTE ADDRESS
     7-  310    PRMERR - PRINT PARAMETER ERROR TEXT
```

Table of contents

```
     4-   53     MACRO DEFINITIONS
     5-  179     $SEREM - UPDATE REMOTE DEFINITION
     6-  240     FINREM - FIND NEXT REMOTE IN SORTED LIST
     7-  264     UPDDLL - UPDATE DLL PARAMETERS
     8-  310     SETxxx - ASSORTED ACTION ROUTINES
     9-  383     DEAFIL-DEALLOCATE A FILE NAME BLOCK
    10-  419     FNDFIL - FIND A MATCHING FILE BLOCK
    11-  450     ALCFIL - ALLOCATE FILE BLOCK
    12-  478     GETBYT - GET SIZE OF ALLOCATION
    12-  495     GETDLL - GET DLL TEMPLATE POINTERS
```

```
                                        .SBTTL  ALCFIL - ALLOCATE FILE BLOCK

                                ;+
                                ; ALCFIL - ALLOCATE FILE BLOCK
                                ;
                                ; INPUTS:       NONE.
                                ;
                                ; OUTPUT:       A FILE BLOCK IS ALLOCATED AND LINKED INTO THE LIST.
                                ;
                                ;               R5 = FILE BLOCK ADDRESS.
                                ;
                                ; ALL OTHER REGISTERS ARE PRESERVED.
                                ;-

 001144  004267  000000G        ALCFIL: JSR     R2,$SAVVR       ; SAVE EM
 001150  012700  000000G                MOV     #$XXAVL,R0      ; POINT TO FREE LIST
 001154  012701  000043                 MOV     #F.LEN,R1       ; LENGTH OF ALLOCATION IN BYTES
 001160  004767  000000G                CALL    $RQBF           ; REQUEST CORE BLOCK
 001164  103406                         BCS     101$            ; IF CS, ERROR
 001166  010005                         MOV     R0,R5           ; COPY BLOCK ADDRESS
 001170  016715  000000G                MOV     $FILHD,(R5)     ; GO TO THE
 001174  010567  000000G                MOV     R5,$FILHD       ;  HEAD OF THE LIST
 001200  000207                         RETURN

 001202                         101$:   ERROR$  #ERRALL         ; ALLOCATION FAILURE
 001212  000167  000000G                JMP     .EXIT           ; THAT'S ALL, FOLKS

```

```
000324  032767  000000G 000000G 80$:    BIT     #E2.BRT,$OPRN2  ; BROADCAST ROUTING TIMER ?
000332  001403                          BEQ     90$             ; IF EQ, NO
000334  016760  000000G 000024          MOV     $EBRT,C.L2TM(R0) ; ELSE STORE IT

000342  032767  000000G 000000G 90$:    BIT     #E2.BEA,$OPRN2  ; BROADCAST ENDNODES ADJACENCIES ?
000350  001403                          BEQ     95$             ; IF EQ, NO
000352  016760  000000G 000016          MOV     $EBEA,C.NBEA(R0) ; ELSE SAVE IT

000360  000241                  95$:    CLC
000362  000207                          RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000364                          101$:   ERROR$  #ERRROU         ; NO ROUTING TEMPLATE
000374  000241                          CLC                     ; DON'T CANCEL REMAINDER OF COMMAND
000376  000207                          RETURN
000400                          102$:   ERROR$  #ERRPNA         ; ROUTING PARAMETERS DON'T APPLY
000410  000261                          SEC                     ; CANCEL WHOLE COMMAND
000412  000207                          RETURN

        000001                          .END
```

```
                                        .SBTTL  SETPTP - SET POINT TO POINT LINE/CIRCUIT STATE
                                ;+
                                ; ** SETPTP - SET POINT TO POINT LINE/CIRCUIT STATE
                                ;
                                ; INPUTS:       R5 = ADDRESS OF SLT TEMPLATE
                                ;               $OPRND  = PARSED OPTIONS
                                ;               $SWICH  = PARSED STATE FLAGS & SW.SLT OPTIONS BITS
                                ;               $HTM    = HELLO TIMER
                                ;
                                ; OUTPUTS:      SLT TEMPLATE UPDATED.
                                ;
                                ; REGISTERS:    NO REGISTERS ARE MODIFIED
                                ;
                                ;-

000570                          SETPTP:
                                ;
                                ; LINE - ALLOW ON/OFF/CLEARED
                                ; CIRCUIT - TREAT CLEARED AS OFF. IF REQUEST IS FOR OFF,
                                ;          AND LINE IS CLEARED DON'T MARK IT FOR LOAD.
                                ;
000570  032767  000000G 000000G         BIT     #OC.STA,$OPRND  ; CHANGE STATE ?
000576  001435                          BEQ     10$             ; IF EQ, DON'T DO IT
000600  022767  000000G 000000G         CMP     #CO.LIN,$COMP   ; IS THIS A LINE ?
000606  001410                          BEQ     5$              ; IF EQ, YES
000610  032765  000000G 000000G         BIT     #LF.MFL,S$LFLG(R5) ; LINE CLEARED ?
000616  001004                          BNE     5$              ; IF NE, NO
000620  032767  000000G 000000G         BIT     #SW.ENA,$SWICH  ; REQUESTED STATE = OFF ?
000626  001421                          BEQ     10$             ; IF EQ, YES - DON'T MARK IT FOR LOAD
000630  042765  000000C 000000G 5$:     BIC     #LF.ENA!LF.MFL,S$LFLG(R5) ; ASSUME CLEARED
000636  032767  000000G 000000G         BIT     #SW.LOA,$SWICH  ; NEW STATE = CLEARED ?
000644  001412                          BEQ     10$             ; IF EQ, YES
000646  052765  000000G 000000G         BIS     #LF.MFL,S$LFLG(R5) ; ELSE ASSUME OFF
000654  032767  000000G 000000G         BIT     #SW.ENA,$SWICH  ; MARK FOR ENABLE ?
000662  001403                          BEQ     10$             ; IF EQ, NO - STATE = OFF
000664  052765  000000G 000000G         BIS     #LF.ENA,S$LFLG(R5) ; ELSE SET LINE STATE = ON

000672  032767  000000G 000000G 10$:    BIT     #OC.HTM,$OPRND  ; HELLO TIMER SPECIFIED ?
000700  001421                          BEQ     27$             ; IF EQ, NO
000702  032765  000000G 000006          BIT     #LF.MTP,C.FLG(R5) ; MULTIPOINT CIRCUIT ?
000710  001407                          BEQ     26$             ; IF EQ, NO
000712  032765  000000G 000000G         BIT     #SL$TYP,C.STS(R5) ; TYPE PRESENT ?
000720  001403                          BEQ     26$             ; IF EQ, NO - MULTIPOINT TRIB
000722  105765  000000G                 TSTB    S$LTYP(R5)      ; CHECK FOR MULTIPOINT CONTROLLER
000726  001006                          BNE     27$             ; IF NE, MULTPOINT CONTROLLER - HTM ON TRIBS
                                ;
                                ; HELLO TIMER
                                ;
000730  016765  000000G 000000G 26$:    MOV     $HTM,S$LHTM(R5) ; ELSE SET HELLO TIMER
000736  052765  000000G 000000G         BIS     #SL$HTM,C.STS(R5) ; REMEMBER IT FOR UPDATE
                                ;
                                ; SERVICE [ENABLED/DISABLED]
                                ;
000744  032767  000000C 000000G 27$:    BIT     #SW.SEN!SW.SDI,$SWICH   ; SERVICE PARSED ?
000752  001412                          BEQ     30$                     ; IF EQ, NOPE
000754  042765  000000G 000000G         BIC     #LF.SER,S$LFLG(R5)      ; ELSE ASSUME SERVICE ENABLED
000762  032767  000000G 000000G         BIT     #SW.SDI,$SWICH          ; DISABLE SERVICE FOR CIRCUIT ?
```

```
                .TITLE  SETSVC - SET SVC PARAMETERS
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1981, 1982, 1983, 1984, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ;       CFE - SET SVC PARAMETERS
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  22-OCT-81
        ;       DECNET-11M/S V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;

```

```
                                                .SBTTL  $SEUNT - SET UNIT PARAMETERS
                                        ;+
                                        ; *** - $SEUNT - SET UNIT PARAMETERS
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       THE UNIT TEMPLATE IS UPDATED
                                        ;
                                        ;-

000134                                  $SEUNT::
000134  004767  000424                          CALL    SETSTA          ; SET ANY STATION PARAMETERS
000140  012700  000000G                         MOV     #$TMLST,R0      ; GET START OF TEMPLATE LIST
                                        ;
                                        ; SEARCH FOR THE DDM
                                        ;
000144  004767  000000G                 10$:    CALL    $FNPRO          ; FIND NEXT PROCESS
000150  103002                                  BCC     15$             ; IF CC, CONTINUE
000152  000167  000404                          JMP     60$             ; ELSE NO MORE
000156  122760  000000G 000000G 15$:    CMPB    #CS.DDM,C.STS(R0) ; IS THIS A DDM TEMPLATE ?
000164  001367                                  BNE     10$             ; IF NE, NO - KEEP LOOKING
000166  016067  000000G 177604                  MOV     C.PRM(R0),DDM   ; SAVE THE DDM PROCESS NAME
000174  105067  000000G                         CLRB    $FLAGS          ; CLEAR THE FLAGS BYTE
000200  116067  000002G 000000G                 MOVB    C.PRM+2(R0),$FLAGS ; SAVE FLAGS BYTE
                                        ;
                                        ; SEARCH FOR THE RIGHT CONTROLLER
                                        ;
000206  011000                          20$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
000210  001002                                  BNE     25$             ; IF NE, MORE
000212  000167  000344                          JMP     60$             ; ELSE END OF TEMPLATES
000216  122760  000000G 000000G 25$:    CMPB    #CS.CNT,C.STS(R0) ; IS THIS A CONTROLLER TEMPLATE ?
000224  001370                                  BNE     20$             ; IF NE, NO .. KEEP LOOKING
000226  032767  000000G 000000G                 BIT     #WC.CNT,$WILD   ; ALL CONTROLLERS ?
000234  001004                                  BNE     30$             ; IF NE, YES
000236  026760  000000G 000000G                 CMP     $CNTNM,C.PRM(R0) ; ELSE, IS THIS THE RIGHT CONTROLLER ?
000244  001360                                  BNE     20$             ; IF NE, NO - KEEP LOOKING
                                        ;
                                        ; SEARCH FOR THE RIGHT UNIT
                                        ;
000246  010001                          30$:    MOV     R0,R1           ; SAVE PREVIOUS TEMPLATE ADDRESS
000250  011000                                  MOV     (R0),R0         ; GET NEXT TEMPLATE
000252  001543                                  BEQ     60$             ; IF EQ, END OF TEMPLATES
000254  122760  000000G 000000G                 CMPB    #CS.UNT,C.STS(R0) ; IS THIS A UNIT TEMPLATE ?
000262  001412                                  BEQ     33$             ; IF EQ, YES
000264  032767  000000G 000000G                 BIT     #WC.CNT,$WILD   ; POSSIBLY MORE CONTROLLERS ?
000272  001533                                  BEQ     60$             ; IF EQ, NO
000274  122760  000000G 000000G                 CMPB    #CS.CNT,C.STS(R0) ; IS IT A CONTROLLER TEMPLATE ?
000302  001761                                  BEQ     30$             ; IF EQ, YES .. GET NEXT UNIT
000304  010100                                  MOV     R1,R0           ; ELSE BACK UP A TEMPLATE ...
000306  000716                                  BR      10$             ; ... MAYBE A DDM
000310  032767  000000G 000000G 33$:    BIT     #FL.MUX,$FLAGS  ; IS THIS A MUX DEVICE ?
000316  001410                                  BEQ     35$             ; IF EQ, NO .. ASSUME RIGHT UNIT
000320  032767  000000G 000000G                 BIT     #WC.UNT,$WILD   ; ALL UNITS ?
000326  001004                                  BNE     35$             ; IF NE, YES
```

```
                                                                                    H 10
SETX2P - SET X.25 LEVEL 2 PARAM MACRO V05.03b Saturday 29-Jun-85 05:24  Page 5-1
Symbol table

CS.X2P= ****** GX       OL.WSZ= ****** GX       X$2HBT= ****** GX       $BSZ  = ****** GX       $RTM   = ****** GX
C.STS = ****** GX       WC.CNT= ****** GX       X$2LIN= ****** GX       $DVNAM= ****** GX       $SEX2P   000000RG
OL.BSZ= ****** GX       WC.LIN= ****** GX       X$2RTC= ****** GX       $HBT  = ****** GX       $TMLST= ****** GX
OL.HBT= ****** GX       WC.UNT= ****** GX       X$2RTT= ****** GX       $OPRND= ****** GX       $WILD  = ****** GX
OL.RTC= ****** GX       X$2BSZ= ****** GX       X$2WSZ= ****** GX       $RTC  = ****** GX       $WSZ   = ****** GX
OL.RTT= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000212    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8289 Words  ( 33 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:04.24
SY:SETX2P.V2,[132,134]SETX2P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETX2P
```

```
                                                                                    I 10
SETX2P      CREATED BY  MACRO  ON 29-JUN-85 AT 05:24       PAGE 1
```

```
**FILE**ID**SHOBJ
```

```
  SSSSSSSS  HH      HH    000000    BBBBBBBB            JJ
  SSSSSSSS  HH      HH    000000    BBBBBBBB            JJ
SS          HH      HH  00      00  BB      BB          JJ
SS          HH      HH  00      00  BB      BB          JJ
SS          HH      HH  00      00  BB      BB          JJ
SS          HH      HH  00      00  BB      BB          JJ
  SSSSSS    HHHHHHHHHH  00      00  BBBBBBBB            JJ
  SSSSSS    HHHHHHHHHH  00      00  BBBBBBBB            JJ
        SS  HH      HH  00      00  BB      BB  JJ      JJ
        SS  HH      HH  00      00  BB      BB  JJ      JJ
        SS  HH      HH  00      00  BB      BB  JJ      JJ      ....
        SS  HH      HH  00      00  BB      BB  JJ      JJ      ....
SSSSSSSS    HH      HH    000000    BBBBBBBB      JJJJJJ        ....
SSSSSSSS    HH      HH    000000    BBBBBBBB      JJJJJJ        ....
```

```
LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

I 11

SHOBJ - LIST OBJECT CHARACTERIS MACRO V05.03b Saturday 29-Jun-85 05:25

```
;****
; LOCAL MACROS
;****

        .MACRO  PRINT   TEXT
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
        .ENDC
        CALL    $PRINT          ; PRINT MESSAGE
        .ENDM   PRINT

        .MACRO  ERROR$  TEXT
;
; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
; MESSAGE STRING.
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
        .ENDC
        CALL    $CFERR                  ; PRINT ERROR MESSAGE
        .ENDM   ERROR$

        .MACRO  EPRINT  TEXT
;
; PRINT TEXT ON ERROR LUN
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
        .ENDC
        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
        .ENDM   EPRINT
;
; GENERATE A STRING
;
        .MACRO  STRING  NAME,TEXT
NAME:   .ASCII  /TEXT/
        NAME'L = . - NAME
        .ENDM
;
; DEFINE A PROTOCOL STRING
;
        .MACRO  PRO     A
        .NCHR   $$$,<A>
        .PSECT  TEXT,D
$$$1 = .
        .ASCII  /A/
        .PSECT
        .WORD   $$$,$$$1
        .ENDM


;****
; LIBRARY MACRO CALLS
;****

        .MCALL  CHADF$,SVCDF$,CALLR

```

```
    673 004122  105024                          CLRB    (R4)+                 ; CLEAR VALUE
    674 004124  032765  000000G 000000G         BIT     #SV$WSZ,C.STS(R5)     ; WINDOW SIZE SPECIFIED?
    675 004132  001403                          BEQ     50$                   ; IF EQ, NO
    676 004134  016564  000000G 177776          MOV     S$VWSZ(R5),-2(R4)     ; USE TEMPLATE VALUE
    677 004142  010502                  50$:    MOV     R5,R2                 ; COPY POINTER
    678 004144  062702  000000G                 ADD     #S$VRCL,R2            ; POINT TO REMAINDER OF DATA
    679         000002                          .REPT   2
    680                                         MOV     (R2)+,(R4)+           ; MAX RECALLS, RECALL TIMER
    681                                         .ENDR
    682 004154  005024                          CLR     (R4)+                 ; NO HELLO,
    683 004156  005024                          CLR     (R4)+                 ;  (SPARE - USED TO BE LISTEN TIMER)
    684 004160  005024                          CLR     (R4)+                 ;  COST, YET
    685 004162  012724  000003                  MOV     #OFFL,(R4)+           ; ASSUME STATE = OFF
    686 004166  012724  001330'                 MOV     #OFF,(R4)+            ; ...
    687 004172  010546                          MOV     R5,-(SP)              ; PUSH R5
    688 004174  004767  000122                  CALL    GETSTA                ; GET STATION INFO
    689 004200  103420                          BCS     55$                   ; NO STATION TEMPLATE ?
    690 004202  132765  000000G 000000G         BITB    #SF.ENA,S$TFLG(R5)    ; ENABLED ?
    691 004210  001406                          BEQ     53$                   ; IF EQ, NOPE
    692 004212  012764  000002  177774          MOV     #ONL,-4(R4)           ; FIX UP DISPLAY, OTHERWISE
    693 004220  012764  001326' 177776          MOV     #ON,-2(R4)            ;
    694 004226  156564  000000G 177772  53$:    BISB    S$TCST(R5),-6(R4)     ; SET REAL COST,
    695 004234  016564  000000G 177766          MOV     S$THTM(R5),-12(R4)    ; HELLO TIMER,
    696 004242  012605                  55$:    MOV     (SP)+,R5              ; RESTORE R5
    697 004244  105067  174753                  CLRB    SVC01                 ; ASSUME NO DTE NUMBER
    698 004250  032765  000000G 000000G         BIT     #SV$DTE,C.STS(R5)     ; DTE PRESENT ?
    699 004256  001410                          BEQ     60$                   ; IF EQ, NOPE
    700 004260  112767  000054  174735          MOVB    #<',>,SVC01           ; FIX FORMAT STRING
    701         000005                          .REPT   5
    702                                         MOV     (R2)+,(R4)+           ; NUMBER
    703                                         .ENDR
    704                                 ;
    705                                 ; FORMAT AND PRINT SVC PARAMETERS
    706                                 ;
    707 004300  010346                  60$:    MOV     R3,-(SP)              ; SAVE R3
    708 004302  012703  001674'                 MOV     #SVCTBL,R3            ; POINT TO FORMATTING CONTROL TABLE
    709 004306  004767  000726                  CALL    PRINT                 ; PRINT IT
    710 004312  012603                          MOV     (SP)+,R3              ; RESTORE R3
    711 004314  000167  177356                  JMP     10$                   ; CONTINUE
    712
    713 004320  000207                  90$:    RETURN
    714
```

```
**FILE**ID**SHOEXE
```

```
  SSSSSSSS   HH      HH     OOOOOO     EEEEEEEEEE   XX      XX   EEEEEEEEEE
  SSSSSSSS   HH      HH     OOOOOO     EEEEEEEEEE   XX      XX   EEEEEEEEEE
SS           HH      HH   OO      OO   EE           XX      XX   EE
SS           HH      HH   OO      OO   EE           XX      XX   EE
SS           HH      HH   OO      OO   EE             XX  XX     EE
SS           HH      HH   OO      OO   EE             XX  XX     EE
  SSSSSS     HHHHHHHHHH   OO      OO   EEEEEEEE         XX       EEEEEEEE
  SSSSSS     HHHHHHHHHH   OO      OO   EEEEEEEE         XX       EEEEEEEE
        SS   HH      HH   OO      OO   EE             XX  XX     EE
        SS   HH      HH   OO      OO   EE             XX  XX     EE
        SS   HH      HH   OO      OO   EE           XX      XX   EE               ....
        SS   HH      HH   OO      OO   EE           XX      XX   EE               ....
SSSSSSSS     HH      HH     OOOOOO     EEEEEEEEEE   XX      XX   EEEEEEEEEE       ....
SSSSSSSS     HH      HH     OOOOOO     EEEEEEEEEE   XX      XX   EEEEEEEEEE       ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                .SBTTL  SEGSIZ - PRINT ECL SEGMENT SIZE
                                        ;+
                                        ; ** SEGSIZ - PRINT ECL SEGMENT SIZE
                                        ;
                                        ; INPUT:        R4 - ADDRESS OF DEC$DF TEMPLATE
                                        ;               $RDBSZ = SIZE OF RDB
                                        ;
                                        ; OUTPUT:       THE ECL SEGMENT SIZE IS PRINTED.
                                        ;-

003056                                  SEGSIZ:
003056  012701  000613'                         MOV     #ECLSEG,R1       ; POINT TO TEXT STRING
003062  012702  001460'                         MOV     #MSGBLK,R2       ; POINT
003066  016412  000004                          MOV     C.SEG(R4),(R2)   ; PICK UP ECL SEGMENT SIZE
003072  001002                                  BNE     10$              ; IF NE, GOT IT
003074  016712  000000G                         MOV     $RDBSZ,(R2)      ; PICK UP RDB SIZE
003100  000167  177640                  10$:    CALLR   PRINT            ; PRINT IT



                                                .SBTTL  PUTHST FIND AND FORMAT HOST ADDRESS AND NAME

                                        ;+
                                        ;       PUTHST  FIND AND FORMAT HOST ADDRESS AND NAME REMOTE NODE STYLE
                                        ;
                                        ;       INPUT:  R0 = NEXT LOCATION IN PRINTING BUFFER
                                        ;               R2 = DATA BLOCK FOR $EDMSG
                                        ;               R5 = NODE TEMPLATE
                                        ;
                                        ;       OUTPUT: HOST ADDRESS AND NAME ADDED TO PRINTING BUFFER
                                        ;               R1,R2 CORRUPUTED
                                        ;-

003104  016501  000054                  PUTHST: MOV     C.HOS(R5),R1     ; FORMAT HOST ADDRESS
003110  004567  177656                          JSR     R5,PUTADR        ; PUT TWO WORD PHASE 4 ADDRESS
003114  000351' 000406'                         .WORD   HSTMSG,HSTMS2    ; ...
003120  026565  000052  000054                  CMP     C.NUM(R5),C.HOS(R5); HOST AND EXEC THE SAME?
003126  001002                                  BNE     5$               ; NOPE, GET REMOTE
003130  010501                                  MOV     R5,R1            ; COPY NODE TEMPLATE ADDRESS
003132  000414                                  BR      15$              ; AND SKIP ALL THE WORK
003134  012701  000000G                 5$:     MOV     #$TMLST,R1       ; GET START OF TEMPLATE LIST
                                        ;
                                        ; SEARCH FOR THE REMOTE TEMPLATE
                                        ;
003140  011101                          10$:    MOV     (R1),R1          ; GET NEXT TEMPLATE
003142  001423                                  BEQ     20$              ; IF EQ, END OF TEMPLATES JUST RETURN
003144  122761  000000G 000000G                 CMPB    #CS.REM,C.STS(R1) ; IS THIS A REMOTE TEMPLATE ?
003152  001372                                  BNE     10$              ; IF NE, NO .. KEEP LOOKING
003154  026561  000054  000010                  CMP     C.HOS(R5),10(R1) ;CHECK TEMPLATE ADDRESS AGAINST HOST
003162  001366                                  BNE     10$              ; IF NE, NOT THIS ONE, KEEP LOOKING
                                        ;
                                        ; DISPLAY THE REMOTE CHARACTERISICS
                                        ;

003164  012702  001460'                 15$:    MOV     #MSGBLK,R2       ; POINT AT ARGUMENT BLOCK
003170  016112  000004                          MOV     C.NAM(R1),(R2)   ; STORE THE REMOTE NAME
003174  016162  000006  000002                  MOV     C.NAM+2(R1),2(R2);      ...
```

```
    283 001504                          X2PBL3: .BLKW   2                       ;  ...
```

```
                                                .SBTTL  $SEOBJ - UPDATE OBJECT DEFINITION
                                        ;+
                                        ; *** - $SEOBJ - UPDATE OBJECT DEFINITION
                                        ;
                                        ; INPUT:
                                        ;       $OPRND = BIT MASK OF THE OPERANDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       THE OBJECT TEMPLATE IS MODIFIED
                                        ;
                                        ;-

000252                                  $SEOBJ::
000252  012700  000000G                         MOV     #$TMLST,R0      ; GET START OF TEMPLATES
                                        ;
                                        ; SEARCH FOR THE OBJECT TEMPLATE
                                        ;
000256  011000                          10$:    MOV     (R0),R0         ; GET NEXT TEMPLATE
000260  001512                                  BEQ     50$             ; IF EQ, NOT IN LIST .. MAKE NEW ENTRY
000262  122760  000000G 000000G                 CMPB    #CS.OBJ,C.STS(R0) ; IS THIS AN OBJECT TEMPLATE ?
000270  001372                                  BNE     10$             ; IF NE, NO .. KEEP LOOKING
000272  126760  000000G 000004                  CMPB    $TNUM,C.TYP(R0) ; IS THIS THE RIGHT TEMPLATE ?
000300  001366                                  BNE     10$             ; IF NE, NO .. KEEP LOOKING
                                        ;
                                        ; FOUND THE OBJECT DEFINITION
                                        ;
000302  004767  000550                          CALL    CHKMUL          ; CHECK FOR MULTIPLE COPY TASKNAME
000306  103522                                  BCS     57$             ; IF CS, ERROR
000310  032767  000000G 000000G                 BIT     #OO.NAM,$OPRND  ; IS THE NAME FIELD PRESENT ?
000316  001416                                  BEQ     20$             ; IF EQ, NO
000320  005767  000000G                         TST     $TNUM           ; IS THIS OBJECT TYPE ZERO ?
000324  001005                                  BNE     15$             ; IF NE, NO .. OKAY
000326                                          ERROR$  #ERRONA         ; OBJECT TYPE ZERO CANNOT HAVE A TASK NAME
000336  000406                                  BR      20$             ; AND CONTINUE
000340  016760  000000G 000006  15$:            MOV     $TNAM,C.NAM(R0) ; ELSE, STORE NEW TASK NAME
000346  016760  000002G 000010                  MOV     $TNAM+2,C.NAM+2(R0) ; 2ND HALF OF TASK NAME
000354  032767  000000G 000000G 20$:            BIT     #OO.USR,$OPRND  ; IS THE USER FIELD SPECIFIED ?
000362  001412                                  BEQ     30$             ; IF EQ, NO
000364  142760  000100  000005                  BICB    #OF.RLU,C.FLG(R0) ; ASSUME "DEFAULT"
000372  032767  000000G 000000G                 BIT     #SW.RLU,$SWICH  ; IS IT "DEFAULT" ?
000400  001403                                  BEQ     30$             ; IF EQ, YES
000402  152760  000100  000005                  BISB    #OF.RLU,C.FLG(R0) ; IT MUST BE "LOGIN"
000410  032767  000000G 000000G 30$:            BIT     #OO.COP,$OPRND  ; WAS THE NUMBER OF COPIES SPECIFIED ?
000416  001420                                  BEQ     40$             ; IF EQ, NO
000420  142760  000200  000005                  BICB    #OF.SMC,C.FLG(R0) ; ASSUME "SINGLE" COPIES
000426  112760  000001  000012                  MOVB    #1,C.CPY(R0)    ;
000434  032767  000000G 000000G                 BIT     #SW.SMC,$SWICH  ; IS IT "SINGLE" COPIES ?
000442  001406                                  BEQ     40$             ; IF EQ, YES .. OKAY
000444  152760  000200  000005                  BISB    #OF.SMC,C.FLG(R0) ; ELSE, SET TO SPAWN MULTIPLE COPIES
000452  116760  000000G 000012                  MOVB    $COPES,C.CPY(R0) ; STORE THE NUMBER OF COPIES TO SPAWN
000460  032767  000000G 000000G 40$:            BIT     #OO.VFY,$OPRND  ; WAS THE VERIFICATION LEVEL SPECIFIED ?
000466  001535                                  BEQ     80$             ; IF EQ, NO .. LEAVE
000470  142760  000007  000005                  BICB    #7,C.FLG(R0)    ; ELSE, CLEAR OUT THE OLD BITS
000476  156760  000000G 000005                  BISB    $VLVL,C.FLG(R0) ; SET THE VERIFICATION LEVEL
000504  000526                                  BR      80$             ; AND LEAVE
                                        ;
                                        ; CREATE A NEW OBJECT TEMPLATE
```

I 2

```
SETPAR     CREATED BY  MACRO  ON 29-JUN-85 AT 17:50     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00
SYMBOL  VALUE           REFERENCES
SETPAR    000000 RG     #5-73
```

J 2

```
**FILE**ID**SETPRO
```

```
        .TITLE  SETPVC - SET PVC PARAMETERS
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       CFE - SET PVC PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  22-OCT-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                .TITLE  SETREM - SET REMOTE NODE OPTIONS
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1984, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - MODULE TO REPLACE MODIFIED FIELDS IN THE REMOTE TEMPLATE
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  14-DEC-79
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;
```

```
                                        .SBTTL  GETBYT - GET SIZE OF ALLOCATION

                                ;+
                                ; GETBYT - GET SIZE OF ALLOCATION
                                ;
                                ; INPUTS:       R5      = ADDRESS OF FILE BLOCK
                                ;
                                ; OUTPUT:       R0      = SIZE OF ALLOCATION
                                ;-

001216  005046                  GETBYT: CLR     -(SP)              ; CLEAR ACCUMULATED BYTES
001220  116500  000004                  MOVB    F.DEV-1(R5),R0     ; FILENAME STRING LENGTH
001224  010026                          MOV     R0,(SP)+           ; SET ACCUMULATED BYTES, RESTORE STACK
001226  062700  000011                  ADD     #6+3,R0            ; ROUND IT UP TO A
001232  042700  000003                  BIC     #3,R0              ; ..4 BYTE BOUNDARY
001236  000207                          RETURN

                                        .SBTTL  GETDLL - GET DLL TEMPLATE POINTERS

                                ;+
                                ; GETDLL - GET DLL TEMPLATE POINTERS
                                ;
                                ; INPUT:        R0      = ADDRESS OF REMOTE BLOCK
                                ;
                                ; OUTPUT:       SERPT   = ADDRESS OF SER$DF TEMPLATE
                                ;               FILPT   = ADDRESS OF FIL$DF TEMPLATE
                                ;-
001240                          GETDLL: SAVRG   R0                 ; SAVE REMOTE BLOCK ADDRESS
001242  012701  000000G                 MOV     #$TMLST,R1         ; POINT TO LISTHEAD
001246  011101                  10$:    MOV     (R1),R1            ; NEXT TEMPLATE
001250  001422                          BEQ     20$                ; IF EQ, NOT FOUND
001252  126127  000000G 000000G         CMPB    C.STS(R1),#CS.SER ; IS IT A SER$DF ?
001260  001372                          BNE     10$                ; IF NE, NO
001262  026160  000006  000010          CMP     S$ADD(R1),C.ADD(R0) ; FOR CURRENT REMOTE ?
001270  001366                          BNE     10$                ; IF NE, NO
001272  010167  176544                  MOV     R1,SERPT           ; SAVE ADDRESS
001276  011101                          MOV     (R1),R1            ; POINT TO NEXT
001300  010167  176540                  MOV     R1,FILPT           ; SAVE IT
001304  126127  000000G 000000G         CMPB    C.STS(R1),#CS.FIL ; IS IT A FIL$DF ?
001312  001063                          BNE     101$               ; IF NE, TROUBLE
001314  000460                          BR      30$

                                ;
                                ; NO DLL DATABASE: ALLOCATE ONE
                                ;
001316  012700  000000G         20$:    MOV     #$XXAVL,R0         ; POINT TO FREE LIST
001322  012701  000056                  MOV     #S$LEN,R1          ; LENGTH OF ALLOCATION IN BYTES
001326  004767  000000G                 CALL    $RQBF              ; REQUEST CORE BLOCK
001332  103455                          BCS     102$               ; IF CS, ERROR
001334  011601                          MOV     (SP),R1            ; RECOVER REMOTE BLOCK ADDRESS
001336  016160  000010  000006          MOV     C.ADD(R1),S$ADD(R0) ; FILL IN NODE ADDRESS
001344  012760  000001  000004          MOV     #SF.ADD,S$FLG(R0)  ; ONLY ADDRESS, SO FAR
001352  012760  000000G 000000G         MOV     #CS.SER,C.STS(R0) ; MARK IT AS A SER$DF
001360  010067  176456                  MOV     R0,SERPT           ; SAVE ADDRESS
001364  004767  000000G                 CALL    $QINSN             ; LINK SER$DF AFTER REM$DF
001370  062767  000000G 000000G         ADD     #S.LENX,$BYTMN     ; UPDATE RUNTIME BYTE MINIMUM
001376  062767  000000G 000000G         ADD     #S.LENX,$BYTXT     ; AND ACTUAL POOL BYTE EXTENSION
```

```
Symbol table

CS.MOD= ****** GX       C.L2TM  000024          E2.BEA= ****** GX       OE.RTM= ****** GX       $MAXH = ****** GX
CS.ROU= ****** GX       C.NA    000014          E2.BRT= ****** GX       $CFERR= ****** GX       $NN   = ****** GX
C.ACST  000022          C.NBEA  000016          OE.ACO= ****** GX       $EACOS= ****** GX       $OPRND= ****** GX
C.ADD   000004          C.RTM   000012          OE.AHO= ****** GX       $EAHOP= ****** GX       $OPRN2= ****** GX
C.AHOP  000020          C.STS = ****** GX       OE.AMA= ****** GX       $EAMAX= ****** GX       $RTM  = ****** GX
C.CST   000006          ERRPNA  000036R         OE.MXC= ****** GX       $EBEA = ****** GX       $SEROU  000110RG
C.HOP   000010          ERRROU  000000R         OE.MXH= ****** GX       $EBRT = ****** GX       $TMLST= ****** GX
C.LEN = 000026          EX$ERR= 000002          OE.NN = ****** GX       $MAXC = ****** GX

. ABS.  000026    000   (RW,I,GBL,ABS,OVR)
        000414    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8489 Words  ( 34 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.24
SY:SETROU.V2,[132,134]SETROU/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETROU
```

```
000770  001403                          BEQ     30$                     ; NO
000772  052765  000000G 000000G         BIS     #LF.SER,S$LFLG(R5)      ; DISABLE SERVICE FOR CIRCUIT
001000  000207                  30$:    RETURN
```

```
                            ;****
                            ; LOCAL MACROS
                            ;****

                                    .MACRO  PRINT   TEXT
                                    MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                    CALL    $PRINT          ; PRINT MESSAGE
                                    .ENDM   PRINT

                                    .MACRO  ERROR$  TEXT
                            ;
                            ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                            ; MESSAGE STRING.
                            ;
                                    .IF DIF <TEXT><R0>
                                    MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                    .ENDC
                                    CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                    .ENDM   ERROR$

                                    .MACRO  EPRINT  TEXT
                            ;
                            ; PRINT TEXT ON ERROR LUN
                            ;
                                    .IF DIF <TEXT><R0>
                                    MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                    .ENDC
                                    CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                    .ENDM   EPRINT

                                    .MCALL  SVCDF$

000000                              SVCDF$                          ; DEFINE SVC SYMBOLS

                            ;****
                            ; LOCAL DATA
                            ;****

                                    .NLIST  BEX
                                    .ENABL  LC
        000002                      EX$ERR = 2
000000     002    103    106  ERRSVC: .ASCIZ  <EX$ERR>/CFE -- Circuit not in configuration/
000045     002    103    106  ERRUSG: .ASCIZ  <EX$ERR>/CFE -- Usage cannot be permanent/
                                    .EVEN

000110  000000              SCR:    .WORD   0

```

```
000330  026760  000000G 000004          CMP     $UNTNM,C.UNT(R0) ; IS THIS THE RIGHT UNIT ?
000336  001343                          BNE     30$              ; IF NE, NO .. KEEP LOOKING
                                ;
                                ; UPDATE THE UNIT TEMPLATE
                                ;
000340  052760  000000G 000000G 35$:    BIS     #CS.MOD,C.STS(R0) ; INDICATE THAT THIS TEMPLATE WAS MODIFIED
000346  032767  000000G 000000G         BIT     #OC.CST,$OPRND   ; HAS THE COST CHANGED ?
000354  001406                          BEQ     37$              ; IF EQ, NO
000356  016760  000000G 000014          MOV     $COST,C.CST(R0)  ; ELSE, STORE THE NEW COST
000364  052760  000000G 000000G         BIS     #UN$CST,C.STS(R0) ; REMEMBER OPTION
000372  032767  000000G 000000G 37$:    BIT     #OL.PLX,$OPRND   ; HAS THE DUPLEX CHANGED ?
000400  001412                          BEQ     40$              ; IF EQ, NO
000402  042760  000001  000006          BIC     #1,C.CWO(R0)     ; ASSUME FULL DUPLEX
000410  032767  000000G 000000G         BIT     #SW.FUL,$SWICH   ; FULL DUPLEX ?
000416  001003                          BNE     40$              ; IF NE, YES
000420  052760  000001  000006          BIS     #1,C.CWO(R0)     ; SET LINE TO HALF DUPLEX
000426  032767  000000G 000000G 40$:    BIT     #OL.SEC,$OPRND   ; SECONDARY CSR ADDRESS CHANGED ?
000434  001431                          BEQ     50$              ; IF EQ, NO
000436  032767  000000G 000000G         BIT     #FL.LMC,$FLAGS   ; IS THIS A COMIOP DEVICE ?
000444  001410                          BEQ     43$              ; IF EQ, NO .. ERROR
000446  032767  000000G 000000G         BIT     #FL.KMX,$FLAGS   ; IS THIS A KMC MULTIPLEXER ?
000454  001413                          BEQ     45$              ; IF EQ, NO .. OKAY
000456  032760  000007  000004          BIT     #7,C.UNT(R0)     ; IS THIS THE ZERO, NINTH, ... UNIT ?
000464  001407                          BEQ     45$              ; IF EQ, YES .. OKAY
000466  010046                  43$:    MOV     R0,-(SP)         ; SAVE TEMPLATE ADDRESS
000470                                  ERROR$  #ERRSEC          ; NO SECONDARY CSR !!
000500  012600                          MOV     (SP)+,R0         ; RESTORE TEMPLATE ADDRESS
000502  000406                          BR      50$              ; AND SKIP CSR
000504  016760  000000G 000012  45$:    MOV     $SCSR,C.CSR(R0)  ; STORE THE SECONDARY CSR ADDRESS
000512  052760  000000G 000000G         BIS     #UN$SCS,C.STS(R0) ; REMEMBER OPTION
000520  032767  000000G 000000G 50$:    BIT     #OL.DPR,$OPRND   ; HAS THE DPR CHANGED ?
000526  001406                          BEQ     55$              ; IF EQ, NO
000530  016760  000000G 000016          MOV     $DPR,C.DPR(R0)   ; ELSE, STORE THE DPR
000536  052760  000000G 000000G         BIS     #UN$DPR,C.STS(R0) ; REMEMBER OPTION
000544  032767  000000G 000000G 55$:    BIT     #OL.BAU,$OPRND   ; HAS THE BAUD RATE CHANGED ?
000552  001635                          BEQ     30$              ; IF EQ, NO .. LOOK FOR MORE WORK
000554  004767  000134                  CALL    SPEED            ; SET UP THE NEW SPEED
000560  000632                          BR      30$              ; AND LOOK FOR MORE WORK !
000562  000207                  60$:    RETURN
```

I 10

```
SETX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

CS.X2P   = ******  GX       5-108
C.STS    = ******  GX       5-108
OL.BSZ   = ******  GX       5-131
OL.HBT   = ******  GX       5-122
OL.RTC   = ******  GX       5-125
OL.RTT   = ******  GX       5-134
OL.WSZ   = ******  GX       5-128
WC.CNT   = ******  GX       5-114
WC.LIN   = ******  GX       5-110
WC.UNT   = ******  GX       5-118
X$2BSZ   = ******  GX      *5-133
X$2HBT   = ******  GX      *5-124
X$2LIN   = ******  GX       5-112     5-116     5-120
X$2RTC   = ******  GX      *5-127
X$2RTT   = ******  GX      *5-136
X$2WSZ   = ******  GX      *5-130
$BSZ     = ******  GX       5-133
$DVNAM   = ******  GX       5-112     5-116     5-120
$HBT     = ******  GX       5-124
$OPRND   = ******  GX       5-122     5-125     5-128     5-131     5-134
$RTC     = ******  GX       5-127
$RTM     = ******  GX       5-136
$SEX2P     000000 RG       #5-101
$TMLST   = ******  GX       5-102
$WILD    = ******  GX       5-110     5-114     5-118
$WSZ     = ******  GX       5-130
```

J 10

```
SETX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24     PAGE 2
```

Table of contents

5- 169 $SHOBJ - SHOW OBJECT CHARACTERISTICS

```
000000                                         CHADF$                  ; DEFINE THE CHARACTERISTICS WORDS OFFSETS
000000                                         SVCDF$                  ; DEFINE THE SVC FLAGS

                                      ;****
                                      ; LOCAL DATA
                                      ;****

000000                                CURSLT: .BLKW   1                ; CURRENT SLT TEMPLATE ADDRESS
                                      ;
                                      ; DEFINE UNIT TEMPLATE
                                      ;
000000                                        .ASECT
          000000                              .=0
000000                                        .BLKW   1                ; C.LNK
000002                                        .BLKW   1                ; C.STS
000004                                C.UNT:  .BLKW   1                ; UNIT NUMBER
000006                                C.CW0:  .BLKW   1                ; CHARACTERISTICS WORD 0
000010                                C.CW1:  .BLKW   1                ; CHARACTERISTICS WORD 1
000012                                C.CSR:  .BLKW   1                ; SECONDARY CSR ADDRESS
000014                                C.CST:  .BLKW   1                ; LINE COST
000016                                C.DPR:  .BLKW   1                ; DEAD POLLING RATIO
000020                                C.HTM:  .BLKW   1                ; HELLO TIMER
000022                                C.LIM:  .BLKW   1                ; LISTEN TIMER
          000024                      C.LEN=.
000002                                        .PSECT

                                              .NLIST  BEX
                                              .ENABL  LC
          000002                              EX$ERR = 2
                                      ;
                                      ; FORMAT STRINGS - ** NOTE ** EDIT MESSAGE ALWAY MOVES AT LEAST ONE CHARACTER
                                      ;                  FROM THE INPUT STRING INTO THE OUTPUT BUFFER (EVEN IF YOU
                                      ;                  SAY THE INPUT LENGTH IS ZERO FOR A %VA) !!
                                      ;
000002       012      103      151    HEADR:  .ASCIZ  <12>/Circuit characteristics as of %Y %3Z/
000050       012      113      156    HEADR1: .ASCIZ  <12>/Known circuit characteristics as of %Y %3Z/
000124       012      103      151    LINMSG: .ASCII  <12>/Circuit = /
000137                                LINID:  .BLKB   20.
000163       045      126      101    OPTMSG: .ASCIZ  /%VA%VA/
000172       040      040      117    TYPMSG: .ASCIZ  /  Owner = %VA, Type = %VA/
000224       040      040      110    HELMSG: .ASCII  /  Hello timer = %M/
000246       054      040      123    SERMSG: .ASCIZ  /, Service = %VA/
000266       040      040      103    STAMSG: .ASCIZ  /  Cost = %M, State = %VA%VA%VA/
000325       040      040      122    RPRMSG: .ASCIZ  /  Router priority = %M, Maximum routers = %M/
000402       045      062      122    FMT2R5: .ASCIZ  /%2R/                       ; FORMAT NON-DLM PVC NAME
000406       045      061      122    FMTDLM: .ASCIZ  /%1R-%M.%M/                 ; FORMAT DLM STYLE PVC NAME
000420       012      103      151    PVCMSG: .ASCIZ  <12>/Circuit = %VA/
000437       040      040      117    PVCMS2: .ASCIZ      /  Owner = %VA, Type = %VA, DTE = %5R/
000504       040      040      125    PVCMS3: .ASCII      /  Usage = %VA, Channel = %M/
000537       054      040      103    PVC01:  .ASCIZ  /, Counter timer = %M/
000564       040      040      123    PVCMS4: .ASCIZ      /  State = %VA, Maximum data = %M/
000625       040      040      115    PVCMS5: .ASCII      /  Maximum window = %M, Line = /
000663                                PVCLIN: .BLKB   20.
000707       040      040      103    PVCMS6: .ASCIZ      /  Cost = %M, Hello timer = %M/

000745       045      116             SVCM1:  .ASCII  /%N/
000747       103      151      162            .ASCIZ  /Circuit = DLM-%M.%M/
```

```
                                                .SBTTL  GETSTA - GET STATION TEMPLATE
                                        ;+
                                        ; ** GETSTA - GET STATION TEMPLATE
                                        ;
                                        ; INPUTS        $TMLST = TEMPLATE LISTHEAD
                                        ;               $DVNAM = DEVICE, LINE AND TRIB ID
                                        ;
                                        ; OUTPUT        CARRY CLEAR: R5 = ADDRESS OF TEMPLATE
                                        ;               CARRY SET: FAILURE
                                        ;-
004322                                  GETSTA:
004322  016746  000000G                         MOV     $WILD,-(SP)       ; SAVE WILDCARD FLAGS
004326  042767  000000C 000000G                 BIC     #WC.UNT!WC.TRI,$WILD ; TURN OFF WILDCARD PROCESSING
004334  012705  000000G                         MOV     #$TMLST,R5        ; POINT TO START OF TEMPLATES
004340  004767  000000G                         CALL    $FNSLT            ; LOCATE SLT
004344  103402                                  BCS     10$               ; NO SLT ?
004346  004767  000000G                         CALL    $FNSTA            ; TRY TO FIND STATION
004352  012667  000000G                 10$:    MOV     (SP)+,$WILD       ; RESTORE WILDCARD FLAGS
004356  000207                                  RETURN
```

Table of contents

4-   53   MACRO DEFINITIONS
5-   187 $SHEXE - SHOW EXECUTOR CHARACTERISTICS
6-   441   PRINT - PRINT A LINE
7-   460   PUTADR - PUT A PHASE 4 ADDRESS
8-   499   SEGSIZ - PRINT ECL SEGMENT SIZE
8-   519   PUTHST FIND AND FORMAT HOST ADDRESS AND NAME

```
    003202  012701  000424'                 MOV     #HSTNAM,R1      ; POINT AT CONTROL STRING FOR NAME
    003206  004767  000000G                 CALL    $EDMSG          ; FORMAT NAME
    003212  000207                  20$:    RETURN
            000001                          .END
```

```
                                        .SBTTL  $SHLIN - SHOW LINE CHARACTERISTICS
                                ;+
                                ; *** - $SHLIN - SHOW LINE CHARACTERISTICS
                                ;
                                ; INPUT:
                                ;         $WILD = WILD CARDS PRESENT
                                ;
                                ; OUTPUT:
                                ;         THE CHARACTERISTICS FOR THE LINE IS DISPLAYED.
                                ;
                                ;-

001510                          $SHLIN::
001510  012701  000102'                 MOV     #HEADR,R1         ; ASSUME ONLY 1 LINE
001514  032767  000000G 000000G         BIT     #WC.NAM,$WILD     ; KNOWN LINES ?
001522  001402                          BEQ     5$                ; IF EQ, NO .. OKAY
001524  012701  000145'                 MOV     #HEADR1,R1        ; ELSE, DIFFERENT HEADER
001530  012700  000000G         5$:     MOV     #$BUFF,R0         ; POINT TO OUTPUT BUFFER
001534  012702  000000G                 MOV     #$GTBUF,R2        ; POINT AT ARGUMENT BLOCK
001540  004767  000000G                 CALL    $EDMSG            ; FORMAT THE HEADER
001544                                  PRINT   #$BUFF            ; PRINT THE HEADER
001554  012700  000000G                 MOV     #$TMLST,R0        ; GET START OF TEMPLATES
001560  042767  000000G 000000G         BIC     #FL.TRI,$FLAGS    ; NEVER WANT TRIB INFO FOR LINES
                                ;
                                ; SEARCH FOR THE DDM PROCESS NAME
                                ;
001566  004767  000000G         10$:    CALL    $FNPRO            ; FIND THE DDM TEMPLATE
001572  103524                          BCS     40$               ; IF CS, NO MORE
001574  010005                  15$:    MOV     R0,R5             ; SAVE TEMPLATE ADDRESS
001576  122765  000000G 000000G         CMPB    #CS.DDM,C.STS(R5) ; IS IT A DDM TEMPLATE ?
001604  001370                          BNE     10$               ; IF NE, NO .. KEEP LOOKING
001606  026527  000000G 015355          CMP     C.PRM(R5),#^RDLM  ; IS THIS THE DLM DDM ?
001614  001764                          BEQ     10$               ; IF EQ, NOT REALLY A LINE - SKIP IT
001616  016567  000000G 000000G         MOV     C.PRM(R5),$DVNAM  ; STORE THE DDM PROCESS NAME
001624  012767  000000G 176244          MOV     #$TMLST,CURSLT    ; GET NEXT SLT FROM START OF LIST
                                ;
                                ; SEARCH FOR THE RIGHT CONTROLLER
                                ;
001632  011505                  20$:    MOV     (R5),R5           ; GET NEXT TEMPLATE
001634  001503                          BEQ     40$               ; IF EQ, END OF TEMPLATES
001636  122765  000000G 000000G         CMPB    #CS.CNT,C.STS(R5) ; IS THIS A CONTROLLER TEMPLATE ?
001644  001405                          BEQ     24$               ; IF EQ, YES .. OKAY
001646  032767  000000G 000000G 23$:    BIT     #WC.CNT,$WILD     ; ELSE, IS IT ALL CONTROLLERS ?
001654  001766                          BEQ     20$               ; IF EQ, NO .. LOOK FOR CONTROLLER
001656  000426                          BR      30$               ; ELSE, MAYBE A UNIT TEMPLATE
001660  032767  000000G 000000G 24$:    BIT     #WC.CNT,$WILD     ; ALL CONTROLLERS ?
001666  001004                          BNE     25$               ; IF NE, YES
001670  026567  000000G 000000G         CMP     C.PRM(R5),$CNTNM  ; IS THIS THE RIGHT CONTROLLER ?
001676  001355                          BNE     20$               ; IF NE, NO .. KEEP LOOKING
001700  016567  000000G 000000G 25$:    MOV     C.PRM(R5),$CNTNM  ; STORE THE CONTROLLER NUMBER
001706  016567  000002G 177506          MOV     C.PRM+2(R5),CNTBLK+2 ; STORE THE VECTOR ADDRESS
001714  016567  000004G 177476          MOV     C.PRM+4(R5),CNTBLK ; STORE THE CSR ADDRESS
001722  016567  000006G 177474          MOV     C.PRM+6(R5),CNTBLK+4 ; STORE THE PRIORITY
001730  004767  000734                  CALL    GETURM            ; GET THE UNIBUS RUN MASK
                                ;
                                ; LOOK FOR THE UNIT TEMPLATE
                                ;
```

```
                                        ;
000506  005767  000000G         50$:    TST     $TNUM              ; IS THIS OBJECT TYPE ZERO ?
000512  001011                          BNE     52$                ; IF NE, NO - NEED OBJECT NAME
000514  032767  000000G 000000G         BIT     #00.NAM,$OPRND     ; ELSE IS NAME PRESENT ?
000522  001411                          BEQ     55$                ; IF EQ, NO - OK
000524                                  ERROR$  #ERRONA            ; ELSE WARNING
000534  000404                          BR      55$                ; CONTINUE
000536  032767  000000G 000000G 52$:    BIT     #00.NAM,$OPRND     ; IS THE OBJECT TASK NAME PRESENT ?
000544  001521                          BEQ     103$               ; IF EQ, NO - ERROR
000546  005000                  55$:    CLR     R0                 ; INDICATE NO TEMPLATE YET
000550  004767  000302                  CALL    CHKMUL             ; CHECK MULTICOPY TASKNAME
000554  103503                  57$:    BCS     101$               ; IF CS, BAD TASKNAME
000556  012700  000000G                 MOV     #$XXAVL,R0         ; POINT AT END OF TASK LISTHEAD
000562  012701  000013                  MOV     #C.LEN,R1          ; GET LENGTH OF AN OBJECT TEMPLATE
000566  004767  000000G                 CALL    $RQBF              ; TRY TO ALLOCATE A TEMPLATE
000572  103501                          BCS     102$               ; IF CS, ALLOCATION FAILURE
000574  012760  000000G 000000G         MOV     #CS.OBJ,C.STS(R0) ; INDICATE THAT THIS IS AN OBJECT TEMPLATE
000602  116760  000000G 000004          MOVB    $TNUM,C.TYP(R0) ; STORE THE OBJECT TYPE
000610  016760  000000G 000006          MOV     $TNAM,C.NAM(R0) ; STORE THE TASK NAME
000616  016760  000002G 000010          MOV     $TNAM+2,C.NAM+2(R0) ; AND 2ND HALF OF NAME
000624  112760  000002  000005          MOVB    #2,C.FLG(R0)       ; USE VERIFICATION 2, NO FLAGS
000632  112760  000001  000012          MOVB    #1,C.CPY(R0)       ; ASSUME NO COPIES SPECIFIED
000640  032767  000000G 000000G         BIT     #SW.RLU,$SWICH     ; RUN THE TASK UNDER THE LOGIN UIC ?
000646  001403                          BEQ     70$                ; IF EQ, NO .. LEAVE ALONE
000650  152760  000100  000005          BISB    #OF.RLU,C.FLG(R0) ; ELSE, FORCE THE TASK TO RUN UNDER THE LOGIN
000656  032767  000000G 000000G 70$:    BIT     #00.COP,$OPRND     ; WAS THE NUMBER OF COPIES SPECIFIED ?
000664  001412                          BEQ     75$                ; IF EQ, NO
000666  032767  000000G 000000G         BIT     #SW.SMC,$SWICH     ; IS IT ''SINGLE'' COPIES ?
000674  001406                          BEQ     75$                ; IF EQ, YES .  OKAY
000676  152760  000200  000005          BISB    #OF.SMC,C.FLG(R0) ; SET TO SP[illegible] COPIES OF THE TASK
000704  116760  000000G 000012          MOVB    $COPES,C.CPY(R0) ; STORE THE NUMBER OF COPIES TO SPAWN
000712  032767  000000G 000000G 75$:    BIT     #00.VFY,$OPRND     ; WAS THE VERIFICATION SPECIFIED ?
000720  001406                          BEQ     77$                ; IF EQ, NO .. ASSUME LEVEL 2
000722  142760  000007  000005          BICB    #7,C.FLG(R0)       ; ELSE, CLEAR OUT OLD BITS
000730  156760  000000G 000005          BISB    $VLVL,C.FLG(R0)    ; SET THE VERIFICATION LEVEL
000736  004767  000060          77$:    CALL    FINOBJ             ; FIND PLACE FOR TEMPLATE
000742  004767  000000G                 CALL    $QINSN             ; INSERT ENTRY INTO QUEUE
000746  062767  000000G 000000G         ADD     #0.LENX,$BYTMN     ; UPDATE MINIMUM POOL BYTE EXTENSION
000754  062767  000000G 000000G         ADD     #0.LENX,$BYTXT     ; AND CURRENT BYTE EXTENSION
000762  000207                  80$:    RETURN
                                ;
                                ; ERRORS
                                ;
000764                          101$:   ERROR$  #ERRMUL            ; ERROR IN MULTICOPY TASKNAME
000774  000207                          RETURN
000776                          102$:   ERROR$  #ERRBUF            ; ALLOCATION FAILURE
001006  000207                          RETURN
001010                          103$:   ERROR$  #ERRNAM            ; WARNING - OBJECT TASK NAME MUST BE SPECIFIED
001020  000207                          RETURN
```

```
**FILE**ID**SETPRO




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP     RRRRRRRR       OOOOOO
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   PPPPPPPP     RRRRRRRR       OOOOOO
SS           EE               TT       PP      PP   RR      RR   OO      OO
SS           EE               TT       PP      PP   RR      RR   OO      OO
SS           EE               TT       PP      PP   RR      RR   OO      OO
SS           EE               TT       PP      PP   RR      RR   OO      OO
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     RRRRRRRR     OO      OO
  SSSSSS     EEEEEEEE         TT       PPPPPPPP     RRRRRRRR     OO      OO
        SS   EE               TT       PP           RR  RR       OO      OO
        SS   EE               TT       PP           RR  RR       OO      OO
        SS   EE               TT       PP           RR    RR     OO      OO        ....
        SS   EE               TT       PP           RR    RR     OO      OO        ....
SSSSSSSS     EEEEEEEEEE       TT       PP           RR      RR     OOOOOO          ....
SSSSSSSS     EEEEEEEEEE       TT       PP           RR      RR     OOOOOO          ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                        .MCALL  CALLR,SAVRG,RESRG

                                ;****
                                ; LOCAL DATA
                                ;****

                                        .NLIST  BEX
                                        .ENABL  LC
         000002                         EX$ERR = 2
000000      002     103     106 ERRPVC: .ASCIZ  <EX$ERR>/CFE -- Circuit not in configuration/
000045      002     103     106 ERRDTE: .ASCIZ  <EX$ERR>/CFE -- DTE not in configuration/
000106      002     103     106 ERRPAR: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: DTE/
000145      002     103     106 ERRCHN: .ASCIZ  <EX$ERR>/CFE -- Missing parameter: CHANNEL/
000210      002     103     106 ERRPRM: .ASCIZ  <EX$ERR>/CFE -- Zero parameter ignored: %VA for DLM-%M.%M/
000272      115     101     130 TXDAT:  .ASCII  /MAXIMUM DATA/
         000014                         TXDATL = .-TXDAT
000306      115     101     130 TXWND:  .ASCII  /MAXIMUM WINDOW/
         000016                         TXWNDL = .-TXWND
                                        .EVEN
                                        .LIST BE
000324                          FOUND:  .BLKW                   ; FOUND FLAG
000326                          SCR:    .BLKW   10
```

```
                                        .SBTTL  MACRO DEFINITIONS

                                        .MCALL  SAVRG,RESRG

                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                ;****
                                ; LOCAL DATA
                                ;****

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .NLIST  BEX
                                        .ENABL  LC
        000002                          EX$ERR = 2
000000     002     103     106  ERRBUF: .ASCIZ  <EX$ERR>/CFE -- Buffer allocation failure/
                                        .EVEN

                                ;
                                ; DEFINE REMOTE TEMPLATE
                                ;
000000                                  .ASECT
        000000                          .=0
000000                                  .BLKW   1               ; C.LNK
000002                                  .BLKW   1               ; C.STS
000004                          C.NAM:  .BLKW   2               ; REMOTE NODE NAME (RAD50)
000010                          C.ADD:  .BLKW   1               ; REMOTE NODE ADDRESS
        000012                  C.LEN=.
```

```
001404  012700  000000G                 MOV     #$XXAVL,R0        ; POINT TO FREE LIST
001410  012701  000016                  MOV     #F$LEN,R1         ; LENGTH OF ALLOCATION IN BYTES
001414  004767  000000G                 CALL    $RQBF             ; REQUEST CORE BLOCK
001420  103422                          BCS     102$              ; IF CS, ERROR
001422  010067  176416                  MOV     R0,FILPT          ; SAVE ADDRESS
001426  010001                          MOV     R0,R1             ; COPY ADDRESS
001430  012702  000007                  MOV     #F$LEN/2,R2       ; GET LENGTH
001434  005021                  25$:    CLR     (R1)+             ; CLEAR IT..
001436  077202                          SOB     R2,25$            ; ..OUT
001440  012760  000000G 000000G         MOV     #CS.FIL,C.STS(R0) ; MARK IT AS A FIL$DF
001446  017601  000000                  MOV     @(SP),R1          ; GET ADDRESS OF SER$DF
001452  004767  000000G                 CALL    $QINSN            ; LINK IT AFTER SER$DF

001456                          30$:    RESRG   R0
001460  000207                          RETURN

001462  000167  000000G         101$:   JMP     .CONER            ; SOMETHING'S WRONG

001466                          102$:   ERROR$  #ERRALL           ; ALLOCATION FAILURE
001476  000167  000000G                 JMP     .EXIT             ; THAT'S ALL, FOLKS

        000001                          .END
```

```
                                                                     J  6
SETROU     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE             REFERENCES

CS.MOD  = ******  GX        5-138
CS.ROU  = ******  GX        5-134
C.ACST    000022           #4-108      *5-161
C.ADD     000004           #4-101       5-136       5-150       *5-153
C.AHOP    000020           #4-107      *5-165
C.CST     000006           #4-102      *5-146
C.HOP     000010           #4-103      *5-142
C.LEN   = 000026           #4-110
C.L2TM    000024           #4-109      *5-173
C.NA      000014           #4-105      *5-169
C.NBEA    000016           #4-106       5-150      *5-152       *5-177
C.RTM     000012           #4-104      *5-157
C.STS   = ******  GX        5-134      *5-138
ERRPNA    000036 R         #4-92        5-187
ERRROU    000000 R         #4-91        5-184
EX$ERR  = 000002           #4-90        4-91        4-92
E2.BEA  = ******  GX        5-175
E2.BRT  = ******  GX        5-171
OE.ACO  = ******  GX        5-159
OE.AHO  = ******  GX        5-163
OE.AMA  = ******  GX        5-167
OE.MXC  = ******  GX        5-144
OE.MXH  = ******  GX        5-140
OE.NN   = ******  GX        5-148
OE.RTM  = ******  GX        5-155
$CFERR  = ******  GX        5-184       5-187
$EACOS  = ******  GX        5-161
$EAHOP  = ******  GX        5-165
$EAMAX  = ******  GX        5-169
$EBEA   = ******  GX        5-177
$EBRT   = ******  GX        5-173
$MAXC   = ******  GX        5-146
$MAXH   = ******  GX        5-142
$NN     = ******  GX        5-152       5-153
$OPRND  = ******  GX        5-140       5-144       5-148       5-155       5-159       5-163       5-167
$OPRN2  = ******  GX        5-171       5-175
$RTM    = ******  GX        5-157
$SEROU    000110 RG        #5-127
$TMLST  = ******  GX        5-128
```

```
                                                                     K  6
SETROU     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23      PAGE 2
```

```
                                                  .SBTTL SETSTA - SET STATION PARAMETERS
                                          ;+
                                          ; *** - SETSTA - SET STATION PARAMETERS
                                          ;
                                          ; INPUT:
                                          ;       $OPRND = OPERANDS MASK WORD
                                          ;       $SWICH = SWITCH MASK WORD & SW.SLT OPTIONS BITS
                                          ;       $HTM   = HELLO TIMER
                                          ;
                                          ; OUTPUT:
                                          ;       STATION FLAGS ARE UPDATED
                                          ;
                                          ;-

 001002  004567  000000G                  SETSTA: JSR     R5,$SAVRG               ; SAVE REGISTERS
 001006  032767  000000G 000000G                  BIT     #FL.MTP,$FLAGS          ; IS THE CIRCUIT MULTIPOINT ?
 001014  001460                                   BEQ     70$                     ; IF EQ, NO .. LEAVE
                                          ;
                                          ; SEARCH FOR STATIONS
                                          ;
 001016  004767  000000G                  10$:    CALL    $FNSTA                  ; FIND THE NEXT STATION
 001022  103455                                   BCS     70$                     ; IF CS, ALL DONE

 001024  032767  000000C 000000G          20$:    BIT     #OL.STA!OC.STA,$OPRND   ; TRIBUTARY STATE CHANGED ?
 001032  001415                                   BEQ     30$                     ; IF EQ, NO .. LEAVE
 001034  052765  000000G 000000G                  BIS     #CS.MOD,C.STS(R5)       ; INDICATE THAT THIS TEMPLATE IS MOD.
 001042  142765  000000G 000006                   BICB    #SF.ENA,6(R5)           ; ELSE, ASSUME THE CIRCUIT IS NOT ENABLED
 001050  032767  000000G 000000G                  BIT     #SW.ENA,$SWICH          ; IS THE CIRCUIT TO BE ENABLED ?
 001056  001403                                   BEQ     30$                     ; IF EQ, NO
 001060  152765  000000G 000006                   BISB    #SF.ENA,6(R5)           ; ELSE, ENABLE THE STATION

 001066  032767  000000G 000000G          30$:    BIT     #OC.HTM,$OPRND          ; DID HELLO TIMER CHANGE ?
 001074  001411                                   BEQ     40$                     ; IF EQ, NO
 001076  052765  000000G 000000G                  BIS     #CS.MOD,C.STS(R5)       ; INDICATE THAT THIS TEMPLATE IS MOD.
 001104  016765  000000G 000012                   MOV     $HTM,12(R5)             ; ELSE SET IT UP
 001112  052765  000000G 000000G                  BIS     #ST$HTM,C.STS(R5)       ; SET UP FLAGS

 001120  032767  000000C 000000G          40$:    BIT     #SW.SEN!SW.SDI,$SWICH   ; SERVICE PARSED ?
 001126  001412                                   BEQ     50$                     ; IF EQ, NOPE
 001130  142765  000000G 000006                   BICB    #SF.SER,6(R5)           ; ELSE ASSUME SERVICE ENABLED
 001136  032767  000000G 000000G                  BIT     #SW.SDI,$SWICH          ; DISABLE SERVICE FOR CIRCUIT ?
 001144  001403                                   BEQ     50$                     ; NO
 001146  152765  000000G 000006                   BISB    #SF.SER,6(R5)           ; DISABLE SERVICE FOR CIRCUIT

 001154  000720                           50$:    BR      10$                     ; NEXT STATION

 001156  000207                           70$:    RETURN

```

```
                                        .SBTTL  $SESVC - SET SVC PARAMETERS
                                ;+
                                ; *** - $SESVC - SET SVC PARAMETERS
                                ;
                                ; INPUT:
                                ;       $OPRND = OPERANDS PRESENT
                                ;
                                ; OUTPUT:
                                ;       THE SVC TEMPLATE FOR THE PARSED CIRCUIT IS MODIFIED
                                ;
                                ;-

000112                          $SESVC::
000112  022767  000000G 000000G         CMP     #CO.KNC,$COMP   ; KNOWN CIRCUITS ?
000120  001406                          BEQ     5$              ; BR IF YES
000122  032767  000000G 000000G         BIT     #FL.DLM,$FLAGS  ; ELSE SEE IF CICUIT IS DLM
000130  001002                          BNE     5$              ; IF NE, OK
000132  000167  000452                  JMP     80$             ; IF EQ, CAN'T BE A SVC
000136  012700  000000G         5$:     MOV     #$TMLST,R0      ; POINT TO TEMPLATES
000142  005001                          CLR     R1              ; NO TEMPLATES FOUND, YET
000144  032767  000000G 000000G         BIT     #OC.USG,$OPRND  ; USEAGE SPECIFIED ?
000152  001406                          BEQ     10$             ; IF EQ, NO
000154  032767  000000C 000000G         BIT     #SW.INC!SW.OUT,$SWICH ; CHECK FOR PERMANENT
000162  001002                          BNE     10$             ; IF NE, OK
000164  000167  000434                  JMP     102$            ; IF EQ, ERROR
000170  011000                  10$:    MOV     (R0),R0         ; NEXT TEMPLATE
000172  001002                          BNE     12$             ; IF NE, OK
000174  000167  000364                  JMP     75$             ; IF EQ, END OF LIST
000200  122760  000000G 000000G 12$:    CMPB    #CS.SVC,C.STS(R0); IS THIS A SVC ?
000206  001370                          BNE     10$             ; IF NE, NO
000210  022767  000000G 000000G         CMP     #CO.KNC,$COMP   ; KNOWN CIRCUITS ?
000216  001420                          BEQ     15$             ; IF EQ, UPDATE TEMPLATE
000220  032767  000000G 000000G         BIT     #WC.CNT,$WILD   ; WILDCARD CONTROLLERS?
000226  001014                          BNE     15$             ; IF NE, YES
000230  026760  000000G 000000G         CMP     $CNTNM,S$VNAM(R0) ; ELSE COMPARE NAMES
000236  001354                          BNE     10$             ; IF NE, NOT THIS TEMPLATE
000240  032767  000000G 000000G         BIT     #WC.TRI,$WILD   ; WILDCARD TRIBUTARIES?
000246  001004                          BNE     15$             ; IF NE, YES
000250  026760  000000G 000002G         CMP     $STAN,S$VNAM+2(R0) ; COMPARE PART 2
000256  001344                          BNE     10$             ; IF NE, KEEP LOOKING
                                ;
                                ; found one, so process its parameters
                                ;
000260  005201                  15$:    INC     R1              ; INCREMENT FOUND COUNT
000262  032767  000000G 000000G         BIT     #OC.USG,$OPRND  ; USEAGE SPECIFIED ?
000270  001413                          BEQ     30$             ; IF EQ, NO
000272  032767  000000G 000000G         BIT     #SW.INC,$SWICH  ; NEW USAGE = INCOMING ?
000300  001404                          BEQ     20$             ; IF EQ, NO - GOTTA BE OUTGOING
000302  052760  000002  000000G         BIS     #SF$INC,S$VFLG(R0) ; MAKE IT INCOMING
000310  000403                          BR      30$             ; GO PROCESS DTE
000312  042760  000002  000000G 20$:    BIC     #SF$INC,S$VFLG(R0) ; MAKE IT OUTGOING

000320  032767  000000G 000000G 30$:    BIT     #OC.NUM,$OPRND  ; DTE ADDRESS ("NUMBER")?
000326  001415                          BEQ     40$             ; IF EQ, NOPE
000330  052760  000000G 000000G 35$:    BIS     #SV$DTE,C.STS(R0) ; REMEMBER DTE NUMBER PRESENT
000336  012702  000000G                 MOV     #$DTE,R2        ; POINT TO PARSED STRING
000342  010003                          MOV     R0,R3           ; COPY TEMPLATE POINTER
000344  062703  000000G                 ADD     #S$VDTE,R3      ; POINT TO TARGET FIELD
```

```
                                ;+
                                ; *** - SETSTA - SET STATION PARAMETERS, IF ANY
                                ;
                                ; INPUTS:
                                ;       $OPRND = OPERANDS MASK WORD
                                ;
                                ; OUTPUT:
                                ;       STATIONS MODIFIED IF NEEDED
                                ;
                                ;-

000564                          SETSTA:
000564  032767  000000C 000000G         BIT     #<OC.CST!OC.TRI!OC.APR>,$OPRND; ANY STATION INFORMATION PRESENT ?
000572  001447                          BEQ     60$                             ; IF EQ, NO - LEAVE
000574  032767  000000G 000000G         BIT     #OL.SEC,$OPRND                  ; IS IT REALLY STATION INFORMATION?
000602  001043                          BNE     60$                             ; IF NE, NO--MUX MULTIPT CSRS FOR LINE
000604  012700  000000G                 MOV     #$TMLST,R0                      ; POINT TO START OF TEMPLATES
                                ;
                                ; SEARCH FOR STATIONS
                                ;
000610  010005                  10$:    MOV     R0,R5                           ; COPY LAST TEMPLATE ADDRESS
000612  004767  000000G                 CALL    .FNSLT                          ; FIND THE NEXT SLT ADDRESS
000616  103435                          BCS     60$                             ; IF CS, NO MORE
000620  010500                          MOV     R5,R0                           ; SAVE THE TEMPLATE ADDRESS
                                ;
                                ; NOW LOOK FOR STATIONS
                                ;
000622  004767  000000G         20$:    CALL    $FNSTA                          ; FIND NEXT STATION
000626  103770                          BCS     10$                             ; IF CS, NO MORE STATIONS ON THIS LINE

000630  052765  000000G 000000G 30$:    BIS     #CS.MOD,C.STS(R5)               ; ELSE, INDICATE THAT THIS TEMPLATE IS MODIFIED
                                ;
                                ; TRIBUTARY COST
                                ;
000636  032767  000000G 000000G         BIT     #OC.CST,$OPRND                  ; HAS THE COST CHANGED ?
000644  001403                          BEQ     35$                             ; IF EQ, NO
000646  116765  000000G 000007          MOVB    $COST,7(R5)                     ; ELSE, CHANGE THE COST
                                ;
                                ; ACTIVE RATIO
                                ;
000654  032767  000000G 000000G 35$:    BIT     #OC.APR,$OPRND                  ; HAS THE APR CHANGED ?
000662  001403                          BEQ     37$                             ; IF EQ, NO
000664  116765  000000G 000010          MOVB    $APR,10(R5)                     ; ELSE, STORE STATION COST
                                ;
                                ; TRIBUTARY ADDRESS
                                ;
000672  032767  000000G 000000G 37$:    BIT     #OC.TRI,$OPRND                  ; HAS THE TRIBUTARY ADDRESS CHANGED ?
000700  001403                          BEQ     40$                             ; IF EQ, NO
000702  116765  000000G 000005          MOVB    $STADR,5(R5)                    ; ELSE, CHANGE THE TRIBUTARY ADDRESS

000710  000744                  40$:    BR      20$                             ; LOOK FOR NEXT STATION

000712  000207                  60$:    RETURN

```

```
                                                                     J 10
SETX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24      PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME          REFERENCES

EPRINT              #4-76
ERROR$              #4-65
PRINT               #4-60
```

```
                                                                     K 10
**FILE**ID**SETX3P
```

```
		.TITLE	SHOBJ - LIST OBJECT CHARACTERISTICS
		.IDENT	/V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SHOW OBJECT DEFAULTS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;
```

```
000773     040     040     117 SVCM2:  .ASCIZ  /  Owner = Executor, Type = X.25, Usage = %VA/
001050     040     040     115 SVCM3:  .ASCIZ  /  Maximum data = %M, Maximum window = %M/
001121     040     040     115 SVCM4:  .ASCIZ  /  Maximum recalls = %M, Recall timer = %M/
001173     040     040     103 SVCM5:  .ASCII  /  Cost = %M, State = %VA/
001223     054     040     116 SVCO1:  .ASCIZ  /, Number = %5R/

                               ;
                               ; ERROR STRINGS
                               ;
        000002                         EX$ERR = 2
001242     002     103     106 ERRNST: .ASCII  <EX$ERR>/CFE -- No station template for /
001302                         NSTLIN: .BLKB   20.
                               ;
                               ; REPLACEMENT STRINGS
                               ;
001326                                 STRING  ON,     <On>
001330                                 STRING  OFF,    <Off>
001333                                 STRING  EXEC,   <Executor>
001343                                 STRING  NONE,   <(none)>
001351                                 STRING  X25,    <X.25>
001355                                 STRING  SERV,   <X25-Server>
001367                                 STRING  PERM,   <Permanent>
001400                                 STRING  OUTG,   <Outgoing>
001410                                 STRING  INCM,   <Incoming>

001420                                 STRING  ENAB,   <Enabled>
001427                                 STRING  DSAB,   <Disabled>

001437     015     012     040 TRIB:   .ASCII  <15><12>/  Tributary = /
001457                         TRIVAL: .BLKB   10.
001471     054     040     115 APR:    .ASCII  /, Multi-point Active = /
001520                         APRVAL: .BLKB   10.
001532                         SCRACH: .BLKB   10.                     ; SCRATCH BUFFER
001544                         HWP:    .BLKB   3                       ; DDM NAME IF HARDWARE PROTOCOL
                                       .EVEN

001550                         PROTBL: PRO     <None>
001554  000000  000000                 .WORD   0,0                     ; 1 DDCMP
001560                                 PRO     <Bisync>
001564                                 PRO     <SDLC>
001570                                 PRO     <X.25>
001574                                 PRO     <Ethernet>
001600                                 PRO     <???>
001604  000003  001544'                .WORD   3,HWP                   ; 7 HARDWARE PROTOCOL

001610                         DDCTBL: PRO     <DDCMP Point>
001614                                 PRO     <DDCMP Control>
001620                                 PRO     <DDCMP Tributary>

001624  000172' 001736'        TRNTBL: .WORD   TYPMSG,TYPBLK           ; PRINT THE CIRCUIT TYPE AND OWNER
001630  000224' 001746'                .WORD   HELMSG,HELBLK           ; PRINT THE HELLO AND LISTEN TIMERS
001634  000266' 002030'                .WORD   STAMSG,STABLK           ; STATE
001640  000000                         .WORD   0

001642  000420' 001754'        PVCTBL: .WORD   PVCMSG,PVCBLK           ; PVC FORMATTING TABLE
001646  000437' 001760'                .WORD   PVCMS2,PVCBL2
001652  000504' 002002'                .WORD   PVCMS3,PVCBL3
```

```
                                        .SBTTL  SHOPVC - SHOW PERMANENT VIRTUAL CIRCUIT
                                ;+
                                ; ** SHOPVC - SHOW PERMANENT VIRTUAL CIRCUIT
                                ;
                                ; INPUTS
                                ;       R5 = ADDRESS OF DTE$DF TEMPLATE
                                ;       R4 = ADDRESS OF PVC$DF TEMPLATE
                                ;       R3 = ADDRESS OF X3P$DF TEMPLATE
                                ; OUTPUT
                                ;       SWITCHED VIRTUAL CIRCUIT CHARACTERISTICS DISPLAYED.
                                ;
                                ; REGISTERS
                                ;       R0 - R3 ARE CORRUPTED.
                                ;-


004360                          SHOPVC:
004360  022767  000000G 000000G         CMP     #CO.KNC,$COMP   ; KNOWN CIRCUITS ?
004366  001436                          BEQ     20$             ; IF EQ,YES
004370  026764  000000G 000000G         CMP     $CIRID,P$VNAM(R4) ; ELSE COMPARE NAMES
004376  001010                          BNE     10$             ; IF NE, NO MATCH
004400  032767  000000G 000000G         BIT     #FL.DLM,$FLAGS  ; DATA LINK MAPPING ?
004406  001006                          BNE     15$             ; IF YES, HAVE TO DO WILDCARDS
004410  026764  000002G 000002G         CMP     $CIRID+2,P$VNAM+2(R4) ; ELSE COMPARE HIGH ORDER
004416  001422                          BEQ     20$             ; BR IF THEY MATCH
004420  000167  000612          10$:    JMP     120$
004424  032767  000000G 000000G 15$:    BIT     #WC.CNT,$WILD   ; WILDCARD CONTROLLERS?
004432  001004                          BNE     17$             ; BR IF YES
004434  126764  000000G 000002G         CMPB    $CNTNM,P$VNAM+2(R4) ; ELSE DO TRIBS MATCH ?
004442  001366                          BNE     10$             ; IF NE, NOT THIS ONE
004444  032767  000000G 000000G 17$:    BIT     #WC.TRI,$WILD   ; WILDCARD TRIBUTARIES?
004452  001004                          BNE     20$             ; BR IF YES
004454  126764  000000G 000003G         CMPB    $STAN,P$VNAM+3(R4) ; ELSE, DO TRIBS MATCH ?
004462  001356                          BNE     10$             ; IF NE, NOPE
004464  012767  015355  000000G 20$:    MOV     #^RDLM,$DVNAM   ; ENSURE DDM NAME CORRECT
004472  116467  000002G 000000G         MOVB    P$VNAM+2(R4),$CNTNM ; IN CASE KNOWN CIRCUITS
004500  116467  000003G 000000G         MOVB    P$VNAM+3(R4),$STAN ; SET UP FOR STATION SCAN
004506  012702  001754'                 MOV     #PVCBLK,R2      ; POINT TO ARGUMENT BLOCK
004512  026427  000000G 015355          CMP     P$VOWN(R4),#^RDLM ; DATA LINK MAPPING ?
004520  001037                          BNE     30$             ; IF NE, NO
004522  112767  000054  174007          MOVB    #<',>,PVC01     ; ASSUME NOT DLM
004530  005067  175132                  CLR     PVCSW           ; ...
004534  032767  000000G 000000G         BIT     #FL.DLM,$FLAGS  ; DATA LINK MAPPING SYLE CIRCUIT ID?
004542  001004                          BNE     25$             ; IF NE, YES - OK
004544  032767  000000C 000000G         BIT     #WC.CNT!WC.TRI,$WILD ; ELSE, WILDCARDS ?
004552  001722                          BEQ     10$             ; IF EQ, HE WANTS ''DLM'', NOT ''DLM-0.0''
004554  105067  173757          25$:    CLRB    PVC01           ; DON'T PRINT COUNTER TIMER FOR DLM PVC
004560  012767  000707' 175100          MOV     #PVCMS6,PVCSW   ; PRINT COST, HELLO TIMER FOR DLM
004566  012701  000406'                 MOV     #FMTDLM,R1      ; FORMAT DLM STYLE ID STRING
004572  105067  173450                  CLRB    SERMSG          ; SUPPRESS SERVICE PARAMETER
004576  016422  000000G                 MOV     P$VNAM(R4),(R2)+ ; PROCESS ID
004602  116422  000002G                 MOVB    P$VNAM+2(R4),(R2)+ ; CONTROLLER
004606  105022                          CLRB    (R2)+           ; CLEAR HI BYTE
004610  116422  000003G                 MOVB    P$VNAM+3(R4),(R2)+ ; TRIB NUMBER
004614  105022                          CLRB    (R2)+           ; NIX HI BYTE
004616  000410                          BR      40$             ; JOIN COMMON CODE

```

```
        .TITLE  SHOEXE - LIST EXECUTOR OPTIONS
        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1984, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SHOW EXECUTOR CHARACTERISTICS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;

```

```
                                                                J 15
SHOEXE - LIST EXECUTOR OPTIONS  MACRO V05.03b Saturday 29-Jun-85 05:25  Page 8-2
Symbol table

ADDMSG  000332R           C.LKS   000010            HEADR   000215R           NT$CC = 000016            RETMSG  001352R
ADDMS2  000364R           C.LST   000021            HOSMSG  000434R           NT$CON= 000000            ROUMSG  000646R
ATYP    001442R           C.L2TM  000024            HSTMSG  000351R           NT$CTL= 000000            ROUTMP  001456R
CD.LEN= 000022            C.NA    000014            HSTMS2  000406R           NT$DAT= 000002            RO2MSG  001046R
CS.DEC= ****** GX         C.NAM   000004            HSTNAM  000424R           NT$DC = 000012            SEGSIZ  003056R
CS.FEA= ****** GX         C.NBEA  000016            IDMSG   000264R           NT$DIS= 000014            SUBBA   001154R
CS.LLC= ****** GX         C.NDC   000006            MSGBLK  001460R           NT$DLS= 000006            SUBBA2  001201R
CS.NOD= ****** GX         C.NID   000010            NAMMSG  000314R           NT$ILS= 000010            TEMP    000000R
CS.REM= ****** GX         C.NUM   000052            NF$ACC= 100000            NT$IMS= 000002            TIMMSG  001230R
CS.ROU= ****** GX         C.OUTT  000013            NF$EVT= 040000            NT$INT= 000004            VFYCOM  000564R
CS.THI= ****** GX         C.PRM = ****** GX         NF.ACC= 000001            NT$RET= 000032            VFYMSG  000516R
CS.TLO= ****** GX         C.RETF  000015            NF.BCP= 000040            NT$ROU= 000024            $BUFF = ****** GX
C.ACST  000022            C.RTM   000012            NF.END= 000200            NT$RTR= 000030            $CFERR= ****** GX
C.ADD   000004            C.SEG   000004            NF.EVT= 000002            NT$TSP= 000026            $CNFIG= ****** GX
C.AHOP  000020            C.STS = ****** GX         NF.LLI= 000004            N$$ACK= 000011            $EDMSG= ****** GX
C.CST   000006            DELMSG  001303R           NF.LV2= 000100            N$$HDR= 000007            $GTBUF= ****** GX
C.DELF  000016            ECLSEG  000613R           NF.SLI= 000400            N$$OVR= 000022            $PRINT= ****** GX
C.DELW  000017            ERRDEC  000074R           NF.SMC= 000020            OFF     001432R           $PSIPT= ****** GX
C.HOP   000010            ERRECL  000037R           NF.SWP= 000010            PRINT   002744R           $RDBSZ= ****** GX
C.HOS   000054            ERRNOD  000124R           NM$ARA= 176000            PUTADR  002772R           $SHEXE  001500RG
C.INAC  000014            ERRROU  000157R           NM$NOD= 001777            PUTHST  003104R           $TMLST= ****** GX
C.INCT  000012            ERRXPT  000002R           NT$AKD= 000020            P$STHI= ****** GX         $$$   = 000023R     002
C.IPL   000020            EX$ERR= 000002            NT$AKI= 000022            P$STLO= ****** GX         $$$L  = 000015
C.LEN = 000026

. ABS.  000056    000   (RW,I,GBL,ABS,OVR)
        003214    001   (RW,I,LCL,REL,CON)
$$$STR  000040    002   (RO,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9739 Words  ( 39 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:14.55
SY:SHOEXE.V2,[132,134]SHOEXE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SHOEXE
```

```
                                                                K 15
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26      PAGE 1
```

```
    342 001734  010500                    30$:    MOV     R5,R0             ; SAVE CURRENT TEMPLATE
    343 001736  011505                            MOV     (R5),R5           ; GET NEXT TEMPLATE
    344 001740  001441                            BEQ     40$               ; IF EQ, END OF TEMPLATES
    345 001742  122765  000000G 000000G 32$:    CMPB    #CS.UNT,C.STS(R5) ; IS THIS A UNIT TEMPLATE ?
    346 001750  001411                            BEQ     33$               ; IF EQ, YES
    347 001752  032767  000000G 000000G         BIT     #WC.CNT,$WILD     ; MORE CONTROLLERS ?
    348 001760  001431                            BEQ     40$               ; IF EQ, NO .. DISPLAYED ALREADY
    349 001762  122765  000000G 000000G         CMPB    #CS.CNT,C.STS(R5) ; ELSE, IS IT A CONTROLLER TEMPLATE ?
    350 001770  001743                            BEQ     25$               ; IF EQ, YES
    351 001772  000675                            BR      10$               ; ELSE, GET NEXT DEVICE
    352 001774  032767  000000G 000000G 33$:    BIT     #FL.MUX,$FLAGS    ; IS THIS A MUX DEVICE ?
    353 002002  001410                            BEQ     35$               ; IF EQ, NO .. MUST BE RIGHT UNIT
    354 002004  032767  000000G 000000G         BIT     #WC.UNT,$WILD     ; ALL UNITS ?
    355 002012  001004                            BNE     35$               ; IF NE, YES
    356 002014  026567  000000G 000000G         CMP     C.PRM(R5),$UNTNM  ; ELSE, IS THIS THE RIGHT UNIT ?
    357 002022  001344                            BNE     30$               ; IF NE, NO .. KEEP LOOKING
    358 002024  016567  000000G 000000G 35$:    MOV     C.PRM(R5),$UNTNM  ; STORE THE UNIT NUMBER
    359 002032  004767  000222                    CALL    GETUNT            ; GET THE UNIT CHARACTERISTICS
    360 002036  004767  000004                    CALL    DISPLY            ; DISPLAY THE CHARACTERISTICS
    361 002042  000734                            BR      30$               ; AND LOOK FOR NEXT UNIT
    362
    363 002044  000207                    40$:    RETURN
```

```
                                                        .SBTTL  FINOBJ - FIND NEXT OBJECT IN SORTED LIST
                                                ;+
                                                ; *** - FINOBJ - FIND NEXT OBJECT IN SORTED LIST
                                                ;
                                                ; INPUTS:
                                                ;       R0 = TEMPLATE TO BE INSERTED
                                                ;
                                                ; OUTPUTS:
                                                ;       R1 = ADDRESS OF TEMPLATE TO PRECEED CURRENT TEMPLATE
                                                ;
                                                ;-

001022                                          FINOBJ:
001022  012702  000000G                                 MOV     #$TMLST,R2              ; GET START ADDRESS OF TEMPLATES
001026  010201                                  10$:    MOV     R2,R1                   ; SAVE ADDRESS OF PREVIOUS
001030  011202                                          MOV     (R2),R2                 ; GET NEXT TEMPLATE IN LIST
001032  001410                                          BEQ     20$                     ; IF EQ, END OF LIST
001034  122762  000000G 000000G                         CMPB    #CS.OBJ,C.STS(R2)       ; IS THIS A OBJECT TEMPLATE ?
001042  001371                                          BNE     10$                     ; IF NE, NO .. KEEP LOOKING
001044  126062  000004  000004                          CMPB    C.TYP(R0),C.TYP(R2)     ; ELSE, DOES TEMPLATE GO HERE ?
001052  103365                                          BHIS    10$                     ; IF HIS, NO .. KEEP LOOKING
001054  000207                                  20$:    RETURN                          ; ELSE, OKAY
```

Table of contents

```
    4-   53    MACRO DEFINITIONS
    5-   95    $SEPRO - SET PROCESS PARAMETERS
```

```
                                        .SBTTL  $SEPVC - SET PVC PARAMETERS
                                ;+
                                ; *** - $SEPVC - SET PVC PARAMETERS
                                ;
                                ; INPUT:
                                        $OPRND = OPERANDS PRESENT
                                ;
                                ; OUTPUT:
                                        THE PVC TEMPLATE FOR THE PARSED CIRCUIT IS MODIFIED
                                ;       IF THERE IS NO PVC TEMPLATE, ONE IS ALLOCATED.
                                ;
                                ;-

000346                          $SEPVC::
                                ;
                                ; ALLOCATE A PVC TEMPLATE IF NECESSARY
                                ;
000346  005067  177752                  CLR     FOUND              ; NO TEMPLATES FOUND, YET
000352  032767  000000G 000000G         BIT     #FL.CRE,$FLAGS     ; CREATING A NEW PVC ?
000360  001453                          BEQ     10$                ; IF EQ, NO
000362  032767  000000G 000000G         BIT     #OC.DTE,$OPRND     ; CHECK FOR DTE ADDRESS IN COMMAND LINE
000370  001002                          BNE     1$                 ; IF NE, OK
000372  000167  000526                  JMP     103$               ; ELSE ERROR
000376  032767  000000G 000000G 1$:     BIT     #OC.CHN,$OPRND     ; CHECK FOR CHANNEL NUMBER IN COMMAND LINE
000404  001002                          BNE     2$                 ; IF NE, OK
000406  000167  000504                  JMP     100$               ; ELSE ERROR
000412  012700  000000G         2$:     MOV     #$XXAVL,R0         ; POINT TO FREE LIST
000416  012701  000000G                 MOV     #P$VLEN,R1         ; LENGTH OF PVC TEMPLATE
000422  004767  000000G                 CALL    $RQBF              ; REQUEST CORE BLOCK
000426  103002                          BCC     3$                 ; IF CC, OK
000430  000167  000476                  JMP     104$               ; ELSE ALLOCATION FAILURE
000434  062767  000000C 000000G 3$:     ADD     #C$LENX+X$LENX,$BYTMN ; UPDATE BYTE MINIMUM
000442  062767  000000C 000000G         ADD     #C$LENX+X$LENX,$BYTXT ; UPDATE BYTE ALLOCATION
000450  010001                          MOV     R0,R1              ; COPY ADDRESS
000452  005721                          TST     (R1)+              ; SKIP LINK WORD
000454  012721  000000G                 MOV     #CS.PVC,(R1)+      ; MARK AS PVC TEMPLATE
000460  016721  000000G                 MOV     $CIRID,(R1)+       ; STUFF CIRCUIT ID
000464  016721  000002G                 MOV     $CIRID+2,(R1)+     ;  ...
        000006                          .REPT   6
                                        CLR     (R1)+              ; CLEAR REMAINDER
                                        .ENDR
000504  005001                          CLR     R1                 ; NOTHING TO UNLINK FOR DTE CHANGE
000506  000452                          BR      20$                ; JOIN COMMON CODE
                                ;
                                ; LOOK FOR PVC TEMPLATE
                                ;
000510  012700  000000G         10$:    MOV     #$PSIPT,R0         ; POINT TO PSI TEMPLATES
000514  010001                  15$:    MOV     R0,R1              ; SAVE PREVIOUS ADDRESS
000516  011000                          MOV     (R0),R0            ; NEXT TEMPLATE
000520  001561                          BEQ     75$                ; IF EQ, END OF LIST
000522  122760  000000G 000000G         CMPB    #CS.PVC,C.STS(R0); IS THIS A PVC ?
000530  001371                          BNE     15$                ; IF NE, NO
000532  022767  000000G 000000G         CMP     #CO.KNC,$COMP      ; KNOWN CIRCUITS ?
000540  001435                          BEQ     20$                ; IF EQ, UPDATE TEMPLATE
000542  026760  000000G 000000G         CMP     $CIRID,P$VNAM(R0) ; ELSE COMPARE NAMES
000550  001361                          BNE     15$                ; IF NE, NOT THIS TEMPLATE
000552  032767  000000G 000000G         BIT     #FL.DLM,$FLAGS     ; DATA LINK MAPPING ?
000560  001005                          BNE     17$                ; IF NE, YES - GO PROCESS WILDCARDS
```

```
                                                                         K 4
SETREM - SET REMOTE NODE OPTION MACRO V05.03b Monday 01-Jul-85 16:37  Page 4-1
MACRO DEFINITIONS

    110 000042                                  .PSECT
    111                                 ;
    112                                 ; DEFINE SER$DF TEMPLATE
    113                                 ;
    114 000012                                  .ASECT
    115         000000                          . = 0
    116 000000                                  .BLKW   1               ; C.LNK
    117 000002                                  .BLKW   1               ; C.FLG
    118 000004                          S$FLG:  .BLKW   1               ; option flags
    119 000006                          S$ADD:  .BLKW   1               ; Target node address   (16 bit node address)
    120 000010                          S$HST:  .BLKW   1               ; Host node address     (16 bit node address)
    121 000012                          S$HAD:  .BLKW   4               ; Hardware address      (RAD50)
    122 000022                          S$CIR:  .BLKW   4               ; Service circuit       (extended RAD50)
    123 000032                          S$DEV:  .BLKW   1               ; Service device        (extended RAD50)
    124 000034                          S$PSS:  .BLKW   6               ; Service password      (RAD50)
    125 000050                          S$DPA:  .BLKW   2               ; Dump address          (octal binary)
    126 000054                          S$DPC:  .BLKW   1               ; Dump count            (in k bytes)
    127         000056                  S$LEN=.
    128 000042                                  .PSECT
    129                                 ;
    130                                 ; Flags definitions:
    131                                 ;
    132         000001                  SF.ADD = 1                      ; Target address specified
    133         000002                  SF.HST = 2                      ; Host address specified
    134         000004                  SF.HAD = 4                      ; Hardware address specified
    135         000010                  SF.CIR = 10                     ; Service circuit specified
    136         000020                  SF.DEV = 20                     ; Service device specified
    137         000040                  SF.PSS = 40                     ; Service password specified
    138         000100                  SF.DPA = 100                    ; Dump address specified
    139         000200                  SF.DPC = 200                    ; Dump count specified
    140         000400                  SF.PH3 = 400                    ; Service node type is Phase 3
    141
    142                                 ;
    143                                 ; Define FIL$DF template
    144                                 ;
    145 000056                                  .ASECT
    146         000000                          . = 0
    147 000000                                  .BLKW   1               ; Link
    148 000002                                  .BLKW   1               ; Flags
    149 000004                          F$SEC:  .BLKW   1               ; Secondary loader file name
    150 000006                          F$TER:  .BLKW   1               ; Tertiary loader file name
    151 000010                          F$LD:   .BLKW   1               ; Load file name
    152 000012                          F$DIA:  .BLKW   1               ; Diagnostics file name
    153 000014                          F$DUM:  .BLKW   1               ; Dump file name
    154         000016                  F$LEN=.
    155 000042                                  .PSECT
    156
    157                                 ;
    158                                 ; Define file descriptor offsets
    159                                 ;
    160
    161         000036                          MXFSP   = 30.           ; Max filespec length
    162
    163 000016                                  .ASECT
    164         000000                          . = 0
    165 000000                          F.LNK:  .BLKW   1               ; Link word
    166 000002                          F.CNT:  .BLKW   1               ; Use count
```

```
                                                                         L 4
SETREM - SET REMOTE NODE OPTION MACRO V05.03b Monday 01-Jul-85 16:37  Page 4-2
SETREM - SET REMOTE NODE OPTION MACRO V05.03b Monday 01-Jul-85 16:37  Page 12-2
Symbol table

ALCFIL  001144R      F$SEC   000004       RN.SPA= ****** GX   S$DPA   000050       $RNDIA= ****** GX
CS.DEC= ****** GX    F$TER   000006       RN.TLD= ****** GX   S$DPC   000054       $RNDMP= ****** GX
CS.FIL= ****** GX    F.CNT   000002       RN.XXX= ****** GX   S$FLG   000004       $RNDPA= ****** GX
CS.REM= ****** GX    F.DEV   000005       R.LENX= ****** GX   S$HAD   000012       $RNHAR= ****** GX
CS.SER= ****** GX    F.LEN = 000043       SERPT   000042R     S$HST   000010       $RNHDS= ****** GX
C.ADD   000010       F.LNK   000000       SETBIT  000662R     S$LEN = 000056       $RNLDD= ****** GX
C.LEN = 000012       GETBYT  001216R      SETFIL  000702R     S$PSS   000034       $RNSCI= ****** GX
C.NAM   000004       GETDLL  001240R      SETWDR  000640R     S.LENX= ****** GX    $RNSDV= ****** GX
C.STS = ****** GX    MXFSP = 000036       SF.ADD= 000001      UPDDLL  000332R      $RNSLO= ****** GX
DEAFIL  001000R      RN.DCT= ****** GX    SF.CIR= 000010      $BYTMN= ****** GX    $RNSPA= ****** GX
DTBL    000402R      RN.DIA= ****** GX    SF.DEV= 000020      $BYTXT= ****** GX    $RNTLD= ****** GX
ERRALL= ****** GX    RN.DMP= ****** GX    SF.DPA= 000100      $CFERR= ****** GX    $RQBF = ****** GX
ERRBUF  000000R      RN.DPA= ****** GX    SF.DPC= 000200      $FILHD= ****** GX    $SAVAL= ****** GX
EX$ERR= 000002       RN.HAR= ****** GX    SF.HAD= 000004      $NODAD= ****** GX    $SAVVR= ****** GX
FILPT   000044R      RN.HOS= ****** GX    SF.HST= 000002      $NDDNA= ****** GX    $SEREM  000046RG
FINREM  000276R      RN.LDD= ****** GX    SF.PH3= 000400      $NTADD= ****** GX    $SWICH= ****** GX
FNDFIL  001074R      RN.NAM= ****** GX    SF.PSS= 000040      $DPRND= ****** GX    $TMLST= ****** GX
F$DIA   000012       RN.SCI= ****** GX    S$ADD   000006      $QINSN= ****** GX    $XXAVL= ****** GX
F$DUM   000014       RN.SDV= ****** GX    S$CIR   000022      $RLBF = ****** GX    .CONER= ****** GX
F$LD    000010       RN.SLO= ****** GX    S$DEV   000032      $RNDCP= ****** GX    .EXIT = ****** GX
F$LEN = 000016       RN.SNV= ****** GX

. ABS.  000056    000   (RW,I,GBL,ABS,OVR)
        001502    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8546 Words  ( 34 Pages)
Size of core pool: 9992 Words  ( 38 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:15.55
DB2:SETREM.T3,[132,134]SETREM/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETREM
```

```
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37      PAGE 1
```

K 6

```
SETROU     CREATED BY  MACRO  ON 29-JUN-85 AT 05:23     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME        REFERENCES

EPRINT            #4-74
ERROR$            #4-63       5-184      5-187
PRINT             #4-58
```

L 6

```
**FILE**ID**SETSLT
```

```
                                                .SBTTL  FNDACT - FIND TRIBUTARY ACTIVITY
                                        ;+
                                        ; *** - FNDACT - FIND TRIBUTARY ACTIVITY
                                        ;
                                        ; INPUT:
                                        ;       R5 = CURRENT SLT ADDRESS
                                        ;
                                        ; OUTPUT:
                                        ;       C-BIT CLEAR - NO TRIBUTARY ACTIVITY
                                        ;       C-BIT SET - TRIBUTARY ACTIVITY
                                        ;       R0 = DESTROYED
                                        ;
                                        ;-

001160                                  FNDACT:
001160  032767  000000G 000000G                 BIT     #FL.MTP,$FLAGS          ; IS THE LINE MULTIPOINT ?
001166  001415                                  BEQ     20$                     ; IF EQ, NO .. LEAVE
001170  010500                                  MOV     R5,R0                   ; COPY TEMPLATE ADDRESS
                                        ;
                                        ; SEARCH FOR STATIONS
                                        ;
001172  011000                          10$:    MOV     (R0),R0                 ; GET NEXT TEMPLATE
001174  001412                                  BEQ     20$                     ; IF EQ, END OF TEMPLATES

001176  122760  000000G 000000G                 CMPB    #CS.STA,C.STS(R0)       ; IS THIS A STATION TEMPLATE ?
001204  001006                                  BNE     20$                     ; IF NE, NO .. LEAVE
001206  132760  000000G 000006                  BITB    #SF.ENA,6(R0)           ; IS THIS STATION 'ON' ?
001214  001766                                  BEQ     10$                     ; IF EQ, NO
001216  000261                                  SEC                             ; ELSE, INDICATE ACTIVITY
001220  000401                                  BR      30$                     ; AND LEAVE

001222  000241                          20$:    CLC                             ; NO ACTIVITY !
001224  000207                          30$:    RETURN

```

```
        000005                          .REPT   5
                                        MOV     (R2)+,(R3)+       ; MOVE DTE ADDRESS
                                        .ENDR
                                ;
                                ; PARAMETERS APPLICABLE TO BOTH INCOMING AND
                                ; OUTGOING SVCS
                                ;
000362  032767  000000G 000000G 40$:    BIT     #OC.MRC,$OPRND    ; MAX RECALLS ?
000370  001403                          BEQ     45$               ; IF EQ, NO
000372  016760  000000G 000000G         MOV     $MRC,S$VRCL(R0)   ; ELSE UPDATE

000400  032767  000000G 000000G 45$:    BIT     #OC.RTM,$OPRND    ; RECALL TIMER ?
000406  001403                          BEQ     50$               ; IF EQ, NO
000410  016760  000000G 000000G         MOV     $RTM,S$VTIM(R0)   ; ELSE UPDATE

000416  032767  000000G 000000G 50$:    BIT     #OC.MBL,$OPRND    ; MAXIMUM DATA?
000424  001406                          BEQ     55$               ; BR IF NO
000426  016760  000000G 000000G         MOV     $MBL,S$VPSZ(R0)   ; STUFF MAX BLOCKS
000434  052760  000000G 000000G         BIS     #SV$PSZ,C.STS(R0) ; REMEMBER OPTION

000442  032767  000000G 000000G 55$:    BIT     #OC.MWI,$OPRND    ; MAXIMUM WINDOW?
000450  001406                          BEQ     60$               ; BR IF NO
000452  016760  000000G 000000G         MOV     $MWI,S$VWSZ(R0)   ; STUFF VALUE INTO TEMPLATE
000460  052760  000000G 000000G         BIS     #SV$WSZ,C.STS(R0) ; REMEMBER OPTION

000466  032767  000000G 000000G 60$:    BIT     #OC.STA,$OPRND    ; STATE ?
000474  001431                          BEQ     70$               ; BR IF NO
000476  012705  000000G                 MOV     #$TMLST,R5        ; POINT TO LISTHEAD
000502  004767  000000G                 CALL    $FNSLT            ; LOOK FOR SLT
000506  103416                          BCS     65$               ; IF CS, ERROR
000510  004767  000000G                 CALL    $FNSTA            ; LOOK FOR STA$DF
000514  103413                          BCS     65$               ; IF CS, ERROR
000516  142765  000000G 000000G         BICB    #SF.ENA,S$TFLG(R5) ; ASSUME NOT ENABLED
000524  032767  000000G 000000G         BIT     #SW.ENA,$SWICH    ; MARK FOR ENABLE ?
000532  001412                          BEQ     70$               ; IF EQ, NO
000534  152765  000000G 000000G         BISB    #SF.ENA,S$TFLG(R5) ; SET STATE ON
000542  000406                          BR      70$               ; CONTINUE
000544  010046                  65$:    MOV     R0,-(SP)          ; PUSH R0
000546                                  ERROR$  #ERRCIS           ; CIRCUIT NOT FOUND
000556  012600                          MOV     (SP)+,R0          ; RECOVER POINTER

000560  000167  177404          70$:    JMP     10$               ; CONTINUE

000564  022767  000000G 000000G 75$:    CMP     #CO.KNC,$COMP     ; KNOWN CIRCUITS ?
000572  001406                          BEQ     80$               ; IF EQ, YES
000574  032767  000000G 000000G         BIT     #WC.CNT,$WILD     ; WILDCARD CONTROLLERS ?
000602  001002                          BNE     80$               ; IF NE, MAY BE OTHER CIRCUITS FOUND
000604  005701                          TST     R1                ; ELSE WAS IT FOUND ?
000606  001401                          BEQ     101$              ; IF EQ, NO - ERROR

000610  000207                  80$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000612                          101$:   ERROR$  #ERRSVC           ; SVC NOT FOUND
000622  000207                          RETURN
```

```
                                        ;+
                                        ; *** - SPEED - CHANGE THE CURRENT BAUD RATE
                                        ;
                                        ; INPUT:
                                        ;       R0 = ADDRESS OF UNIT TEMPLATE
                                        ;       $BAUD = NEW BAUD RATE
                                        ;
                                        ; OUTPUT:
                                        ;       C.CW0 = UPDATED
                                        ;
                                        ;-

000714  012701  000036                  SPEED:  MOV     #TBLEN+2,R1     ; ASSUME HIGH SPEED LINE
000720  026727  000000G 022600                  CMP     $BAUD,#9600.    ; HIGH SPEED LINE ?
000726  101016                                  BHI     20$             ; BR IF YES
000730  005001                                  CLR     R1              ; ELSE CLEAR INDEX POINTER
000732  026761  000000G 000076' 10$:            CMP     $BAUD,RATETB(R1) ; BAUD RATE IN TABLE ?
000740  001411                                  BEQ     20$             ; IF EQ, YES .. OKAY
000742  005721                                  TST     (R1)+           ; ELSE, SKIP TO NEXT ENTRY
000744  020127  000034                          CMP     R1,#TBLEN       ; AT END OF TABLE ?
000750  101770                                  BLOS    10$             ; IF LOS, NO .. KEEP LOOKING
000752                                          ERROR$  #ERRBAU         ; ELSE, ILLEGAL BAUD RATE
000762  000415                                  BR      30$             ; AND LEAVE
000764  006201                          20$:    ASR     R1              ; GET INDEX / 2
000766  000301                                  SWAB    R1              ; GET TRANSMIT SPEED IN R1
000770  042760  177400  000006                  BIC     #177400,C.CW0(R0) ; CLEAR CURRENT TRANS & RECV SPEEDS
000776  050160  000006                          BIS     R1,C.CW0(R0)    ; SET NEW TRANSMIT SPEED
        000004                                  .REPT   4
                                                ASL     R1              ; GET RECEIVE SPEED IN R1
                                                .ENDR
001012  050160  000006                          BIS     R1,C.CW0(R0)    ; SET NEW RECEIVE SPEED
001016  000207                          30$:    RETURN                  ; AND RETURN TO CALLER

        000001                                  .END
```

```
**FILE**ID**SETX3P




  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   XX      XX     333333     PPPPPPPP
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT   XX      XX     333333     PPPPPPPP
SS           EE               TT       XX      XX   33      33   PP      PP
SS           EE               TT       XX      XX   33      33   PP      PP
SS           EE               TT         XX  XX             33   PP      PP
SS           EE               TT         XX  XX             33   PP      PP
  SSSSSS     EEEEEEEE         TT           XX             33     PPPPPPPP
  SSSSSS     EEEEEEEE         TT           XX             33     PPPPPPPP
        SS   EE               TT         XX  XX             33   PP
        SS   EE               TT         XX  XX             33   PP
        SS   EE               TT       XX      XX   33      33   PP                ....
        SS   EE               TT       XX      XX   33      33   PP                ....
SSSSSSSS     EEEEEEEEEE       TT       XX      XX     333333     PP                ....
SSSSSSSS     EEEEEEEEEE       TT       XX      XX     333333     PP                ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                ;****
                                ; LIBRARY MACRO CALLS
                                ;****

                                        .MCALL  OBJDF$

000000                                  OBJDF$                  ; DEFINE OBJECT BIT DEFINITIONS

                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        .IF DIF <TEXT>,<R0>
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        .ENDC
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                ;****
                                ; LOCAL DATA
                                ;****

                                ;
                                ; DEFINE OBJECT TEMPLATE
                                ;
000000                                  .ASECT
                000000                  .=0
000000                                  .BLKW   1               ; C.LNK
000002                                  .BLKW   1               ; C.STS
000004                          C.TYP:  .BLKB   1               ; OBJECT TYPE
000005                          C.FLG:  .BLKB   1               ; OBJECT FLAG BYTE
000006                          C.NAM:  .BLKW   2               ; OBJECT TASK NAME (RAD50)
000012                          C.CPY:  .BLKB   1               ; NUMBER OF COPIES TO SPAWN
                000013          C.LEN=.
000000                                  .PSECT
```

```
001656  000564' 002012'                 .WORD   PVCMS4,PVCBL4
001662  000625' 002020'                 .WORD   PVCMS5,PVCBL5
001666  000707' 002022'         PVCSW:  .WORD   PVCMS6,PVCBL6
001672  000000                          .WORD   0

001674  000745' 001726'         SVCTBL: .WORD   SVCM1,SVCBLK            ; SVC FORMATTING TABLE
001700  000773' 001732'                 .WORD   SVCM2,SVCBLK+4
001704  001050' 001736'                 .WORD   SVCM3,SVCBLK+10
001710  001121' 001742'                 .WORD   SVCM4,SVCBLK+14
001714  000224' 001746'                 .WORD   HELMSG,SVCBLK+20
001720  001173' 001752'                 .WORD   SVCM5,SVCBLK+24
001724  000000                          .WORD   0

001726                          SVCBLK: .BLKW   4                       ; USAGE MAX RECALL, RECALL TIMER
001736                          TYPBLK: .BLKW   4                       ; TYPE, OWNER
001746                          HELBLK: .BLKW   3                       ; HELLO TIMER, SERVICE STRING DESCR.

001754                          PVCBLK: .BLKW   2                       ; PVC PARAMETER BLOCK
001760                          PVCBL2: .BLKW   9.
002002                          PVCBL3: .BLKW   4
002012                          PVCBL4: .BLKW   3
002020                          PVCBL5: .BLKW   1
002022                          PVCBL6: .BLKW   3.

002030                          STABLK: .BLKW   4                       ; LOADING ARGUMENT BLOCK
002040   001437'                        .WORD   TRIB                    ; TRIBUTARY ADDRESS STRING
002042                                  .BLKW   2                       ; LENGTH AND ADDRESS OF APR
```

```
    794 004620  012701  000402'         30$:    MOV     #FMT2R5,R1         ; POINT TO FORMAT STRING
    795 004624  016422  000000G                 MOV     P$VNAM(R4),(R2)+ ; FIRST PART OF NAME
    796 004630  016422  000002G                 MOV     P$VNAM+2(R4),(R2)+ ; NEXT PART OF NAME
    797 004634  105067  173406                  CLRB    SERMSG             ; SUPPRESS SERVICE PARAMETER
    798
    799 004640  012702  001754'         40$:    MOV     #PVCBLK,R2         ; POINT TO ARGUMENT BLOCK
    800 004644  012700  000137'                 MOV     #LINID,R0          ; POINT TO BUFFER
    801 004650  004767  000000G                 CALL    $EDMSG             ; FORMAT CIRCUIT ID
    802 004654  162700  000137'                 SUB     #LINID,R0          ; CALCULATE STRING LENGTH
    803 004660  012702  001754'                 MOV     #PVCBLK,R2         ; POINT TO ARG BLOCK
    804 004664  010022                          MOV     R0,(R2)+           ; LOAD UP STRING LENGTH
    805 004666  012722  000137'                 MOV     #LINID,(R2)+       ; POINT TO STRING
    806 004672  012722  000006                  MOV     #NONEL,(R2)+       ; ASSUME NO OWNER
    807 004676  012722  001343'                 MOV     #NONE,(R2)+        ; ...
    808 004702  026427  000000G 015355          CMP     P$VOWN(R4),#^RDLM ; CHECK FOR DLM
    809 004710  001006                          BNE     50$                ; IF NE, NOT DLM
    810 004712  012762  000010  177774          MOV     #EXECL,-4(R2)      ; ELSE OWNER = EXECUTOR
    811 004720  012762  001333' 177776          MOV     #EXEC,-2(R2)       ; ...
    812 004726  012722  000004          50$:    MOV     #X25L,(R2)+        ; TYPE IS X25
    813 004732  012722  001351'                 MOV     #X25,(R2)+         ;  ...
    814                                 ;
    815                                 ; DTE ADDRESS AND USAGE
    816                                 ;
    817 004736  010500                          MOV     R5,R0              ; POINT TO DTE TEMPLATE
    818 004740  062700  000000G                 ADD     #D$TADR,R0         ; POINT TO DTE ADDRESS FIELD
    819         000005                          .REPT   5
    820                                         MOV     (R0)+,(R2)+        ; MOVE DTE ADDRESS
    821                                         .ENDR
    822 004756  012722  000011                  MOV     #PERML,(R2)+       ; USAGE IS PERMANENT
    823 004762  012722  001367'                 MOV     #PERM,(R2)+        ; ...
    824 004766  016422  000000G                 MOV     P$VLCN(R4),(R2)+; CHANNEL NUMBER
    825 004772  016422  000000G                 MOV     P$VCT(R4),(R2)+; COUNTER TIMER
    826                                 ;
    827                                 ; STATE, MAXIMUM DATA, MAXIMUM WINDOW
    828                                 ;
    829 004776  012722  000002                  MOV     #ONL,(R2)+         ; ASSUME STATE = ON
    830 005002  012722  001326'                 MOV     #ON,(R2)+          ; ...
    831 005006  105764  000000G                 TSTB    P$VSTA(R4)         ; TEST STATE
    832 005012  001006                          BNE     60$                ; IF NE, ON
    833 005014  012762  000003  177774          MOV     #OFFL,-4(R2)       ; ELSE
    834 005022  012762  001330' 177776          MOV     #OFF,-2(R2)        ; STATE = OFF
    835 005030  016322  000000G         60$:    MOV     X$3DFP(R3),(R2)+; ASSUME DEFAULT DATA SIZE
    836 005034  032764  000000G 000000G         BIT     #PV$PSZ,C.STS(R4) ; MAX DATA ON PVC$DF ?
    837 005042  001403                          BEQ     70$                ; IF EQ, NOT SPECIFIED
    838 005044  016462  000000G 177776          MOV     P$VPSZ(R4),-2(R2): BLOCK SIZE
    839 005052  116322  000000G         70$:    MOVB    X$3DW(R3),(R2)+ ; ASSUME DEFAULT WINDOW
    840 005056  105022                          CLRB    (R2)+              ; CLEAR HI BYTE
    841 005060  032764  000000G 000000G         BIT     #PV$WSZ,C.STS(R4) ; MAX WINDOW ON PVC$DF ?
    842 005066  001403                          BEQ     80$                ; IF EQ, NO
    843 005070  016462  000000G 177776          MOV     P$VWSZ(R4),-2(R2) ; WINDOW SIZE
    844                                 ;
    845                                 ; FORMAT LINE ID
    846                                 ;
    847 005076  010346                  80$:    MOV     R3,-(SP)           ; PUSH R3
    848 005100  012700  000663'                 MOV     #PVCLIN,R0         ; POINT TO LINE BUFFER
    849 005104  010503                          MOV     R5,R3              ; POINT TO
    850 005106  062703  000000G                 ADD     #D$TLIN,R3         ;  LINE ID
```

```
                                                                K 14
SHOEXE - LIST EXECUTOR OPTIONS  MACRO V05.03b Saturday 29-Jun-85 05:25  Page 4
MACRO DEFINITIONS

     53                                         .SBTTL  MACRO DEFINITIONS
     54
     55                                 ;
     56                                 ; LIBRARY MACRO CALLS
     57                                 ;
     58
     59                                         .MCALL  NSFDF$,MSGDF$,DECTP$
     60
     61 000000                                  DECTP$                  ; DEFINE DEC$DF TEMPLATE OFFSETS
     62 000000                                  NSFDF$                  ; DEFINE NSP FEATURES MASK BITS
     63 000000                                  MSGDF$                  ; DEFINE NSP MESSAGE SYMBOLS
     64
     65                                 ;
     66                                 ; LOCAL MACROS
     67                                 ;
     68
     69                                         .MACRO  PRINT   TEXT
     70                                         MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
     71                                         CALL    $PRINT          ; PRINT MESSAGE
     72                                         .ENDM   PRINT
     73
     74                                         .MACRO  ERROR$  TEXT
     75                                 ;
     76                                 ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
     77                                 ; MESSAGE STRING.
     78                                 ;
     79                                         .IF DIF <TEXT><R0>
     80                                         MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
     81                                         .ENDC
     82                                         CALL    $CFERR                  ; PRINT ERROR MESSAGE
     83                                         .ENDM   ERROR$
     84
     85                                         .MACRO  EPRINT  TEXT
     86                                 ;
     87                                 ; PRINT TEXT ON ERROR LUN
     88                                 ;
     89                                         .IF DIF <TEXT><R0>
     90                                         MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
     91                                         .ENDC
     92                                         CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
     93                                         .ENDM   EPRINT
     94
     95                                         .MACRO  STRING  TEXT
     96                                         .SAVE
     97                                         .PSECT  $$$STR, D ,RO, CON, LCL
     98                                         $$$ = .
     99                                         .ASCII  /TEXT/
    100                                         $$$L = . - $$$
    101                                         .RESTORE
    102                                         .WORD   $$$L,$$$
    103                                         .ENDM
    104
    105                                 ;
    106                                 ; LOCAL DATA
    107                                 ;
    108
    109                                 ;
```

```
                                                                L 14
SHOEXE - LIST EXECUTOR OPTIONS  MACRO V05.03b Saturday 29-Jun-85 05:25  Page 4-1
MACRO DEFINITIONS
```

```
                                                                                  K 15
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26       PAGE 1

SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL   VALUE                REFERENCES

ADDMSG     000332 R           #4-155      5-291
ADDMS2     000364 R           #4-157      5-291
ATYP       001442 R           #4-178      5-350
CD.LEN   = 000022             #4-61
CS.DEC   = ******   GX          5-267
CS.FEA   = ******   GX          5-256
CS.LLC   = ******   GX          5-235     5-246
CS.NOD   = ******   GX          5-226
CS.REM   = ******   GX          8-545
CS.ROU   = ******   GX          5-215
CS.THI   = ******   GK          5-410     5-420
CS.TLO   = ******   GX          5-410     5-415
C.ACST     000022             #4-139      5-401
C.ADD      000004             #4-132      5-371
C.AHOP     000020             #4-138      5-400
C.CST      000006             #4-133      5-384
C.DELF     000016             #4-61       5-319
C.DELW     000017             #4-61       5-328
C.HOP      000010             #4-134      5-385
C.HOS      000054             #4-120      8-532       8-535      8-547
C.INAC     000014             #4-61       5-342
C.INCT     000012             #4-61       5-300
C.IPL      000020             #4-61
C.LEN    = 000026             #4-121     #4-141
C.LKS      000010             #4-61       5-308
C.LST      000021             #4-61
C.L2TM     000024             #4-140      5-388
C.NA       000014             #4-136      5-399
C.NAM      000004             #4-116      5-284       5-285      8-554      8-555
C.NBEA     000016             #4-137      5-387
C.NDC      000006             #4-61       5-309
C.NID      000010             #4-117      5-274       5-276
C.NUM      000052             #4-119      5-289       8-535
C.OUTT     000013             #4-61       5-301
C.PRM    = ******   GX          5-237     5-248       5-258      5-362      5-364
C.RETF     000015             #4-61       5-343
C.RTM      000012             #4-135      5-386
C.SEG      000004             #4-61       8-512
C.STS    = ******   GX          5-215     5-226       5-235      5-246      5-256      5-267      5-410      5-415      5-420
                                8-545
DELMSG     001303 R           #4-172      5-334
ECLSEG     000613 R           #4-163      8-510
ERRDEC     000074 R           #4-149      5-438
ERRECL     000037 R           #4-148      5-436
ERRNOD     000124 R           #4-150      5-430
ERRROU     000157 R           #4-151      5-434
ERRXPT     000002 R           #4-147      5-432
EX$ERR   = 000002             #4-146      4-147       4-148      4-149      4-150      4-151
HEADR      000215 R           #4-152      5-200
HOSMSG     000434 R           #4-160      5-310
HSTMSG     000351 R           #4-156      8-534
HSTMS2     000406 R           #4-158      8-534
```

```
                                                                                  L 15
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26       PAGE 2
```

```
                                          .SBTTL  DISPLY - DISPLAY LINE CHARACTERISTICS
                                  ;+
                                  ; *** - DISPLY - DISPLAY LINE CHARACTERISTICS
                                  ;
                                  ; INPUT:
                                  ;       CNTBLK = CONTROLLER INFORMATION
                                  ;       UNTBLK = UNIT INFORMATION
                                  ;       OPTBLK = OPTIONAL INFORMATION
                                  ;       X2PBLK = X.25 LEVEL 2 INFORMATION
                                  ;
                                  ; OUTPUT:
                                  ;       CHARACTERISTICS ARE DISPLAYED
                                  ;-

002046                            DISPLY:
002046  004767  000730                    CALL    GETSLT          ; GET INITIAL SLT/STATION INFORMATION
002052  103475                            BCS     101$            ; IF CS, NO SLT MACRO
002054  004767  001346                    CALL    FMTLIN          ; FORMAT THE LINE ID
002060                                    PRINT   #LINMSG         ; PRINT THE LINE ID
002070  012703  001344'                   MOV     #TBL1,R3        ; POINT R3 AT ARGUMENT BLOCK TABLE
002074  012704  001364'                   MOV     #TBL2,R4        ; POINT R4 AT INPUT STRING TABLE

002100  012700  000000G           10$:    MOV     #$BUFF,R0       ; GET THE OUTPUT BUFFER
002104  012301                            MOV     (R3)+,R1        ; GET THE INPUT BUFFER
002106  001413                            BEQ     20$             ; IF EQ, END OF LIST
002110  012402                            MOV     (R4)+,R2        ; GET THE ARGUMENT BLOCK
002112  032467  175762                    BIT     (R4)+,DSPOPT    ; OPTIONAL INFO SELECTED ?
002116  001770                            BEQ     10$             ; IF EQ, NO
002120  004767  000000G                   CALL    $EDMSG          ; EDIT THE LINE
002124                                    PRINT   #$BUFF          ; PRINT THE LINE
002134  000761                            BR      10$             ; REPEAT
002136  032767  000000G 000000G   20$:    BIT     #FL.MTP,$FLAGS  ; IS THE LINE MULTI-POINT ?
002144  001001                            BNE     30$             ; IF NE, YES
002146  000436                            BR      50$             ; AND LEAVE
002150                            30$:
002150  032767  000000G 000000G           BIT     #FL.TRI,$FLAGS  ; DOES USER WANT TRIBUTARY INFO. ?
002156  001425                            BEQ     45$             ; IF EQ, NO
002160  052767  000000G 000000G           BIS     #FL.TRB,$FLAGS  ; SIGNAL TRIBUTARY ADDRESS REQUIRED
002166  004767  001234                    CALL    FMTLIN          ; FORMAT THE LINE ID
002172                                    PRINT   #LINMSG         ; PRINT LINE ID
002202  012700  000000G           40$:    MOV     #$BUFF,R0       ; POINT AT OUTPUT BUFFER
002206  012701  000631'                   MOV     #STAMSG,R1      ; POINT AT INPUT BUFFER
002212  012702  001456'                   MOV     #STABLK,R2      ; POINT AT ARGUMENT BLOCK
002216  004767  000000G                   CALL    $EDMSG          ; FORMAT THE STRING
002222                                    PRINT   #$BUFF          ; PRINT THE LINE
002232  004736                    45$:    CALL    @(SP)+          ; BACK TO GETSLT AGAIN
002234  103345                            BCC     30$             ; IF CC, GOT NEW LINE
002236  042767  000000G 000000G           BIC     #FL.TRB,$FLAGS  ; ELSE, CLEAR TRIBUTARY ADDRESS BIT
002244  000207                    50$:    RETURN
                                  ;
                                  ; ERROR CONDITIONS
                                  ;
002246                            101$:   ERROR$  #ERRLIS         ; LINE NOT IN SYSTEM
002256  000207                            RETURN
```

```
                                                 .SBTTL  CHKMUL - CHECK FOR MULTICOPY TASKNAME
                                        ;+
                                        ; INPUTS         $TNAM = PARSED TASK NAME
                                        ;                R0 = OBJECT TEMPLATE (0 IF NONE)
                                        ;
                                        ; OUTPUT         CARRY CLEAR:
                                        ;                 SINGLE COPY NAME OK FOR SINGLE COPY OBJECT OR
                                        ;                 MULTICOPY NAME OK FOR MULTICOPY OBJECT
                                        ;                CARRY SET:
                                        ;                 WRONG NAME FOR OBJECT TYPE
                                        ;
                                        ;                NOTE: IT IS VALID TO HAVE A MULTICOPY TASK WITH
                                        ;                      MAXIMUM COPIES OF 1
                                        ;
                                        ; REGISTERS      ALL REGISTERS PRESERVED.
                                        ;-

 001056  004267  000000G                CHKMUL: JSR     R2,$SAVVR          ; SAVE VOLATILE REGISTERS
 001062  012702  000000G                        MOV     #$TNAM,R2          ; POINT TO PARSED NAME
 001066  032767  000000G 000000G                BIT     #OO.NAM,$OPRND     ; PARSED OBJECT NAME ?
 001074  001004                                 BNE     10$                ; IF NE, YES
 001076  010002                                 MOV     R0,R2              ; ELSE COPY TEMPLATE POINTER
 001100  001460                                 BEQ     50$                ; IF EQ, CATCH ERROR LATER
 001102  062702  000006                         ADD     #C.NAM,R2          ; POINT TO NAME IN TEMPLATE
                                        ;
                                        ; CHECK OBJECT NAME FOR MULTICOPY FORMAT
                                        ;
 001106  012700  000244'                10$:    MOV     #OBJNAM,R0         ; POINT TO SCRATCH BUFFER
 001112  012201                                 MOV     (R2)+,R1           ; FETCH FIRST PART
 001114  011246                                 MOV     (R2),-(SP)         ; PUSH SECOND PART
 001116  004767  000000G                        CALL    $C5TA              ; CONVERT RAD50 TO ASCII
 001122  012601                                 MOV     (SP)+,R1           ; FETCH NEXT PART
 001124  004767  000000G                        CALL    $C5TA              ; CONVERT RAD50 TO ASCII
 001130  124027  000040                 15$:    CMPB    -(R0),#40          ; SCAN BACK OVER BLANKS
 001134  001775                                 BEQ     15$                ; IF EQ, KEEP ON SCANNIN'
 001136  005200                                 INC     R0                 ; BACKTRACKED ONE TOO MANY
 001140  012702  177774                         MOV     #-4,R2             ; NOT MULTICOPY YET
 001144  005202                         16$:    INC     R2                 ; COUNT NEXT DOLLAR
 001146  124027  000044                         CMPB    -(R0),#'$          ; IS IT DOLLAR ?
 001152  001774                                 BEQ     16$                ; IF EQ, YEP
 001154  005001                                 CLR     R1                 ; NOT MULTIPLE COPIES YET
                                        ;
                                        ; CHECK FOR MULTIPLE COPIES ON COMMAND LINE
                                        ;
 001156  032767  000000G 000000G                BIT     #OO.COP,$OPRND     ; CHECK FOR PARSED COPIES
 001164  001406                                 BEQ     20$                ; BR IF NONE
 001166  032767  000000G 000000G                BIT     #SW.SMC,$SWICH     ; CHECK PARSED FLAGS
 001174  001412                                 BEQ     30$                ; IF EQ, NOT MULTIPLE COPIES
 001176  005201                                 INC     R1                 ; ELSE SET FLAG
 001200  000410                                 BR      30$                ; JOIN COMMON CODE
                                        ;
                                        ; CHECK FOR MULTIPLE COPIES IN TEMPLATE
                                        ;
 001202  016600  000002                 20$:    MOV     2(SP),R0           ; IS THERE A TEMPLATE YET ?
 001206  001405                                 BEQ     30$                ; IF EQ, NO - ASSUME SINGLE COPY
 001210  132760  000200  000005                 BITB    #OF.SMC,C.FLG(R0)  ; SPAWN MULTI COPIES ?
 001216  001401                                 BEQ     30$                ; IF EQ, NO
```

```
                .TITLE  SETPRO - SET PROCESS OPTIONS
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; CFE - SET PROCESS PARAMETERS
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  14-DEC-79
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;

```

```
000562  026760  000002G 000002G          CMP     $CIRID+2,P$VNAM+2(R0) ; COMPARE PART 2
000570  001351                           BNE     15$                   ; IF NE, KEEP LOOKING
000572  000420                           BR      20$                   ; ELSE GOT ONE

000574  032767  000000G 000000G 17$:     BIT     #WC.CNT,$WILD         ; WILDCARD CONTROLLERS ?
000602  001004                           BNE     18$                   ; IF NE, YES
000604  126760  000000G 000002G          CMPB    $CNTNM,P$VNAM+2(R0) ; COMPARE CONTROLLER NUMBERS
000612  001340                           BNE     15$                   ; BR IF WRONG CONTROLLER
000614  032767  000000G 000000G 18$:     BIT     #WC.TRI,$WILD         ; WILDCARD TRIBS ?
000622  001004                           BNE     20$                   ; IF NE, YES
000624  126760  000000G 000003G          CMPB    $STAN,P$VNAM+3(R0) ; COMPARE TRIB NUMBERS
000632  001330                           BNE     15$                   ; IF NE, WRONG TRIB

000634  005267  177464          20$:     INC     FOUND                 ; INCREMENT FOUND COUNT
000640  032767  000000G 000000G          BIT     #OC.CHN,$OPRND        ; CHANNEL NUMBER ?
000646  001403                           BEQ     30$                   ; IF EQ, NOPE
000650  016760  000000G 000000G          MOV     $CHN,P$VLCN(R0)       ; UPDATE CHANNEL NUMBER IN TEMPLATE

000656  032767  000000G 000000G 30$:     BIT     #OC.MBL,$OPRND        ; MAXIMUM BLOCK SPECIFIED ?
000664  001416                           BEQ     40$                   ; IF EQ, NO
000666  005767  000000G                  TST     $MBL                  ; ZERO VALUE ?
000672  001005                           BNE     35$                   ; IF NE, OK FOR PVC
000674  004167  000416                   JSR     R1,PRMERR             ; ELSE SOFT PARAMETER ERROR
000700  000272' 000014                   .WORD   TXDAT,TXDATL          ; PASS ERROR TYPE
000704  000406                           BR      40$                   ; CONTINUE PROCESSING
000706  016760  000000G 000000G 35$:     MOV     $MBL,P$VPSZ(R0)       ; ELSE UPDATE MAX BLOCK
000714  052760  000000G 000000G          BIS     #PV$PSZ,C.STS(R0) ; REMEMBER PACKET SIZE SPECIFIED

000722  032767  000000G 000000G 40$:     BIT     #OC.MWI,$OPRND        ; MAXIMUM WINDOW ?
000730  001416                           BEQ     50$                   ; IF EQ, NO
000732  005767  000000G                  TST     $MBL                  ; ZERO VALUE ?
000736  001005                           BNE     45$                   ; IF NE, OK FOR PVC
000740  004167  000352                   JSR     R1,PRMERR             ; ELSE SOFT PARAMETER ERROR
000744  000306' 000016                   .WORD   TXWND,TXWNDL          ; PASS ERROR TYPE
000750  000406                           BR      50$                   ; CONTINUE PROCESSING
000752  016760  000000G 000000G 45$:     MOV     $MWI,P$VWSZ(R0)       ; ELSE UPDATE WINDOW SIZE
000760  052760  000000G 000000G          BIS     #PV$WSZ,C.STS(R0) ; REMEMBER OPTION

000766  032767  000000G 000000G 50$:     BIT     #OC.CTM,$OPRND        ; COUNTER TIMER ?
000774  001407                           BEQ     55$                   ; IF EQ, NO
000776  026027  000000G 015355           CMP     P$VOWN(R0),#^RDLM ; DATA LINK MAPPING ?
001004  001403                           BEQ     55$                   ; IF EQ, YES - IGNORE FOR NOW
001006  016760  000000G 000000G          MOV     $CT,P$VCT(R0)         ; ELSE UPDATE

001014  032767  000000G 000000G 55$:     BIT     #OC.STA,$OPRND        ; STATE ?
001022  001410                           BEQ     70$                   ; IF EQ, NO
001024  105060  000000G                  CLRB    P$VSTA(R0)            ; ASSUME OFF
001030  032767  000000G 000000G          BIT     #SW.ENA,$SWICH        ; STATE = ON ?
001036  001402                           BEQ     70$                   ; IF EQ, NO
001040  105260  000000G                  INCB    P$VSTA(R0)            ; SET STATE OFF

001044  032767  000000G 000000G 70$:     BIT     #OC.DTE,$OPRND        ; CHANGE DTE ?
001052  001403                           BEQ     72$                   ; IF EQ, NO
001054  004767  000132                   CALL    SETDTE                ; SET NEW DTE
001060  103432                           BCS     110$                  ; IF CS, ERROR
001062  000614                  72$:     BR      15$                   ; ELSE CONTINUE

```

```
    167 000004                      .BLKB   1               ; Filespec string length
    168 000005              F.DEV:  .BLKB   MXFSP           ; Filespec string (ASCII)
    169         000043      F.LEN   =       .
    170 000042                      .PSECT
    171
    172                     ;
    173                     ; LOCAL DATA
    174                     ;
    175 000042              SERPT:  .BLKW   1
    176 000044              FILPT:  .BLKW   1
    177
```

```
                                                                     L  5
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37      PAGE 1
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE              REFERENCES

ALCFIL    001144 R          8-366      #11-464
CS.DEC  = ******   GX       5-195
CS.FIL  = ******   GX      12-516       12-544
CS.REM  = ******   GX       5-202        5-223      6-257
CS.SER  = ******   GX      12-509       12-529
C.ADD     000010           #4-108        5-204     *5-231      6-259      6-259      12-511     12-527
C.LEN   = 000012           #4-109        5-218
C.NAM     000004           #4-107       *5-211     *5-212     *5-224     *5-225
C.STS   = ******   GX       5-195        5-202     *5-223      6-257     12-509      12-516    *12-529    *12-544
DEAFIL    001000 R          8-360       #9-396
DTBL      000402 R          7-277       #7-292
ERRALL  = ******   GX      11-474       12-553
ERRBUF    000000 R         #4-97         5-221
EX$ERR  = 000002           #4-96         4-97
FILPT     000044 R         #4-176        8-357    *12-515    *12-539
FINREM    000276 R          5-232       #6-252
FNDFIL    001074 R          8-362      #10-434
F$DIA     000012           #4-152        7-305
F$DUM     000014           #4-153        7-306
F$LD      000010           #4-151        7-304
F$LEN   = 000016           #4-154       12-536     12-541
F$SEC     000004           #4-149        7-302
F$TER     000006           #4-150        7-303
F.CNT     000002           #4-166       *8-378     *9-398
F.DEV     000005           #4-168        8-371     10-439     10-441      12-489
F.LEN   = 000043           #4-169        8-371      9-408     10-441      11-466
F.LNK     000000           #4-165
GETBYT    001216 R          8-374        9-411    #12-488
GETDLL    001240 R          7-276      #12-505
MXFSP   = 000036           #4-161        4-168
RN.DCT  = ******   GX       7-298
RN.DIA  = ******   GX       7-305
RN.DMP  = ******   GX       7-306
RN.DPA  = ******   GX       7-297
RN.HAR  = ******   GX       7-300
RN.HOS  = ******   GX       7-294
RN.LOD  = ******   GX       7-304
RN.NAM  = ******   GX       5-209
RN.SCI  = ******   GX       7-299
RN.SDV  = ******   GX       7-295
RN.SLO  = ******   GX       7-302
RN.SNV  = ******   GX       7-296
RN.SPA  = ******   GX       7-301
RN.TLO  = ******   GX       7-303
RN.XXX  = ******   GX       7-274
R.LENX  = ******   GX       5-234        5-235
SERPT     000042 R         #4-175        8-323      8-326      8-338     *12-513    *12-530
SETBIT    000662 R          7-296       #8-338
SETFIL    000702 R          7-302        7-303      7-304      7-305       7-306     #8-356
SETWOR    000640 R          7-294        7-295      7-297      7-298       7-299      7-300      7-301     #8-323
SF.ADD  = 000001           #4-132       12-528
SF.CIR  = 000010           #4-135        7-299
```

```
                                                                     M  5
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37      PAGE 2
```

```
**FILE**ID**SETSLT





  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     SSSSSSSS   LL           TTTTTTTTTT
  SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT     SSSSSSSS   LL           TTTTTTTTTT
SS           EE               TT       SS           LL               TT
SS           EE               TT       SS           LL               TT
SS           EE               TT       SS           LL               TT
SS           EE               TT       SS           LL               TT
  SSSSSS     EEEEEEEE         TT         SSSSSS     LL               TT
  SSSSSS     EEEEEEEE         TT         SSSSSS     LL               TT
        SS   EE               TT               SS   LL               TT
        SS   EE               TT               SS   LL               TT
        SS   EE               TT               SS   LL               TT            ....
        SS   EE               TT               SS   LL               TT            ....
SSSSSSSS     EEEEEEEEEE       TT       SSSSSSSS     LLLLLLLLLL       TT            ....
SSSSSSSS     EEEEEEEEEE       TT       SSSSSSSS     LLLLLLLLLL       TT            ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                  .SBTTL  ACTERR - MULTIPOINT TRIBS ACTIVE ERROR
                                          ;+
                                          ; **     ACTERR - MULTIPOINT TRIBS ACTIVE ERROR
                                          ;
                                          ; INPUT:         R5 = SLT TEMPLATE ADDRESS
                                          ;
                                          ; OUTPUT:        ERROR MESSAGE FORMATTED AND PRINTED
                                          ;
                                          ;
                                          ;-

001226                                    ACTERR:
001226  004267  000000G                           JSR     R2,$SAVVR               ; SAVE VOLATILE REGISTERS
001232  012700  000000G                           MOV     #$BUFF,R0               ; POINT TO OUTPUT BUFFER
001236  012701  000006'                           MOV     #ERRACT,R1              ; POINT TO FORMATTING STRING
001242  112120                                    MOVB    (R1)+,(R0)+             ; MOVE EXIT STATUS BYTE
001244  012702  000000'                           MOV     #ARGLST,R2              ; POINT TO ARGUMENT LIST
001250  016522  000004                            MOV     C.NAM(R5),(R2)+         ; GET DDM NAME
001254  116522  000010                            MOVB    C.CNT(R5),(R2)+         ; CONTROLLER NUMBER
001260  105022                                    CLRB    (R2)+                   ; CLEAR HI BYTE
001262  105067  176571                            CLRB    ACTOPT                  ; ASSUME NOT MUX
001266  032767  000000G 000000G                   BIT     #FL.MUX,$FLAGS          ; MUX ?
001274  001406                                    BEQ     10$                     ; IF EQ, NOPE
001276  112767  000055  176553                    MOVB    #'-,ACTOPT              ; ELSE INCLUDE UNIT NUMBER
001304  116522  000011                            MOVB    C.UNT(R5),(R2)+         ; STUFF UNIT NUMBER
001310  105022                                    CLRB    (R2)+                   ; CLEAR HI BYTE
001312  012702  000000'                   10$:    MOV     #ARGLST,R2              ; POINT AGAIN
001316  004767  000000G                           CALL    $EDMSG                  ; FORMAT IT
001322                                            ERROR$  #$BUFF                  ; PRINT IT
001332  000207                                    RETURN
```

```
    215
    216 000624                     102$:   ERROR$  #ERRUSG         ; USAGE CANNOT BE PERMANENT
    217 000634  000207                     RETURN
    218
    219         000001                     .END
```

```
                                                       L 9
SETUNT - SET LINE UNIT OPTIONS  MACRO V05.03b Monday 12-Aug-85 15:02  Page 7-1
Symbol table

CH.DUP= 000001          C.CSR   000012          EX$WAR= 000000          SPEED   000714R         $DPR  = ****** GX
CH.D11= 000100          C.CST   000014          FL.KMX= ****** GX       SW.FUL= ****** GX       $FLAGS= ****** GX
CH.ECH= 000002          C.CWO   000006          FL.LMC= ****** GX       TBLEN = 000034          $FNPRO= ****** GX
CH.MDT= 000020          C.CW1   000010          FL.MUX= ****** GX       UN$CST= ****** GX       $FNSTA= ****** GX
CH.PAR= 000100          C.DPR   000016          OC.APR= ****** GX       UN$DPR= ****** GX       $OPRND= ****** GX
CH.PRM= 000200          C.LEN = 000024          OC.CST= ****** GX       UN$SCS= ****** GX       $SCSR = ****** GX
CH.PRT= 000007          C.PRM = ****** GX       OC.TRI= ****** GX       WC.CNT= ****** GX       $SEUNT  000134RG
CH.STA= 000040          C.STS = ****** GX       OL.BAU= ****** GX       WC.UNT= ****** GX       $STADR= ****** GX
CH.SYN= 000004          C.UNT   000004          OL.DPR= ****** GX       $APR  = ****** GX       $SWICH= ****** GX
CH.TYP= 000010          DDM     000000R         OL.PLX= ****** GX       $BAUD = ****** GX       $TMLST= ****** GX
CS.CNT= ****** GX       END     000132R         OL.SEC= ****** GX       $CFERR= ****** GX       $UNTNM= ****** GX
CS.DDM= ****** GX       ERRBAU  000002R         RATETB  000076R         $CNTNM= ****** GX       $WILD = ****** GX
CS.MOD= ****** GX       ERRSEC  000034R         SETSTA  000564R         $COST = ****** GX       .FNSLT= ****** GX
CS.UNT= ****** GX       EX$ERR= 000002

. ABS.  000024    000   (RW,I,GBL,ABS,OVR)
        001020    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10014 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:13.33
DB2:SETUNT.T64,[132,134]SETUNT/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETUNT
```

```
                                                       M 9
SETUNT     CREATED BY  MACRO  ON 12-AUG-85 AT 15:02     PAGE 1
```

Table of contents

```
     5-   86     LOCAL DATA
     6-   96     $SEX3P - SET X.25 LEVEL 3 PARAMETERS
```

```

                                        .NLIST  BEX
                                        .ENABL  LC
              000002                    EX$ERR = 2
000000           002     103     106    ERROBJ: .ASCIZ  <EX$ERR>/CFE -- Object not in configuration/
000044           012     117     142    HEADR:  .ASCIZ  <12>/Object characteristics as of %Y %3Z/
000111           012     113     156    HEADR1: .ASCIZ  <12>/Known objects characteristics as of %Y %3Z/
000165           012     117     142    O1MSG:  .ASCIZ  <12>/Object = %M/
000202           040     040     116    O2MSG:  .ASCIZ  /  Name = %2R, Verification = %VA/
000243           040     040     125    O3MSG:  .ASCIZ  /  User = %VA, Copies = %VA/
000276           123     151     156    ONE:    .ASCII  /Single/
              000006                            ONEL=.-ONE
000304           114     157     147    LOGIN:  .ASCII  /Login/
              000005                            LOGINL=.-LOGIN
000311           104     145     146    DEFALT: .ASCII  /Default/
              000007                            DFALTL=.-DEFALT
000320           117     156            ON:     .ASCII  /On/
              000002                            ONL=.-ON
000322           117     146     146    OFF:    .ASCII  /Off/
              000003                            OFFL=.-OFF
000325           111     156     163    INSP:   .ASCII  /Inspect/
              000007                            INSPL=.-INSP
000334           077                    WHAT:   .ASCII  /?/
              000001                            WHATL=.-WHAT
                                                .EVEN

000336                                  HFLAG:  .BLKW   1               ; HEADER FLAG
000340                                  MS1BLK: .BLKW   1               ; MESSAGE 1 ARGUMENT BLOCK
000342                                  MS2BLK: .BLKW   4               ; MESSAGE 2 ARGUMENT BLOCK
000352                                  MS3BLK: .BLKW   4               ; MESSAGE 3 ARGUMENT BLOCK
000362                                  BUFR:   .BLKB   10              ; SCRATCH BUFFER
                                        ;
                                        ; DEFINE THE INPUT STRING TABLE
                                        ;
000372  000165'                         TBL1:   .WORD   O1MSG
000374  000202'                                 .WORD   O2MSG
000376  000243'                                 .WORD   O3MSG
000400  000000                                  .WORD   0
                                        ;
                                        ; DEFINE THE ARGUMENT BLOCK TABLE
                                        ;
000402  000340'                         TBL2:   .WORD   MS1BLK
000404  000342'                                 .WORD   MS2BLK
000406  000352'                                 .WORD   MS3BLK
                                        ;
                                        ; DEFINE THE VERIFICATION STATES
                                        ;
000410  000002  000320'                 VTBL:   .WORD   ONL,ON          ; LEVEL 0
000414  000007  000325'                         .WORD   INSPL,INSP      ; LEVEL 1
000420  000003  000322'                         .WORD   OFFL,OFF        ; LEVEL 2
000424  000001  000334'                         .WORD   WHATL,WHAT      ; LEVEL 3
000430  000001  000334'                         .WORD   WHATL,WHAT      ; LEVEL 4
000434  000001  000334'                         .WORD   WHATL,WHAT      ; LEVEL 5
000440  000001  000334'                         .WORD   WHATL,WHAT      ; LEVEL 6
000444  000001  000334'                         .WORD   WHATL,WHAT      ; LEVEL 7

```

```
                                                .SBTTL  $SHCIR - SHOW CIRCUIT CHARACTERISTICS
                                        ;+
                                        ; *** - $SHCIR - SHOW CIRCUIT CHARACTERISTICS
                                        ;
                                        ; INPUT:
                                        ;       $WILD = WILD CARDS PRESENT
                                        ;
                                        ; OUTPUT:
                                        ;       THE CHARACTERISTICS FOR THE CIRCUIT IS DISPLAYED.
                                        ;
                                        ;-
                                        ;
002046                                  $SHCIR::
002046  012701  000002'                         MOV     #HEADR,R1       ; ASSUME ONLY 1 CIRCUIT
002052  032767  000000G 000000G                 BIT     #WC.NAM,$WILD   ; KNOWN CIRCUITS ?
002060  001402                                  BEQ     5$              ; IF EQ, NO .. OKAY
002062  012701  000050'                         MOV     #HEADR1,R1      ; ELSE, DIFFERENT HEADER
002066  012700  000000G                 5$:     MOV     #$BUFF,R0       ; POINT TO OUTPUT BUFFER
002072  012702  000000G                         MOV     #$GTBUF,R2      ; POINT AT ARGUMENT BLOCK
002076  004767  000000G                         CALL    $EDMSG          ; FORMAT THE HEADER
002102                                          PRINT   #$BUFF          ; PRINT THE HEADER
002112  032767  000000G 000000G                 BIT     #FL.X25,$FLAGS  ; PARSED PSI CIRCUIT ID ?
002120  001133                                  BNE     65$             ; IF NE, YES - SKIP TRANSPORT
                                        ;
                                        ; SEARCH FOR THE DDM PROCESS NAME
                                        ;
002122  012700  000000G                         MOV     #$TMLST,R0      ; GET START OF TEMPLATES
002126  004767  000000G                 10$:    CALL    $FNPRO          ; FIND THE DDM TEMPLATE
002132  103526                                  BCS     65$             ; IF CS, NO MORE
                                        ;;;     BIT     #WC.NAM,$WILD   ; IS THIS FOR ALL CIRCUITS ?
                                        ;;;     BEQ     15$             ; IF EQ, NO
002134  032767  000000G 000000G                 BIT     #FL.TRI,$FLAGS  ; USER SPECIFIED A TRIB?
002142  001005                                  BNE     15$             ; YES, LEAVE IT
002144  005067  000000G                         CLR     $STAN           ; MATCH ON FIRST (ONLY?) TRIB
002150  052767  000000G 000000G                 BIS     #FL.TRI,$FLAGS  ; ELSE, INCLUDE TRIBUTARY INFO.
002156  010005                          15$:    MOV     R0,R5           ; SAVE TEMPLATE ADDRESS
002160  026527  000000G 015355                  CMP     C.PRM(R5),#^RDLM ; IS THIS THE DLM DDM TEMPLATE ?
002166  001757                                  BEQ     10$             ; IF EQ, SKIP NOW - SPECIAL CASE LATER
002170  122765  000000G 000000G                 CMPB    #CS.DDM,C.STS(R5) ; IS IT A DDM TEMPLATE ?
002176  001353                                  BNE     10$             ; IF NE, NO .. KEEP LOOKING
002200  016567  000000G 000000G                 MOV     C.PRM(R5),$DVNAM ; STORE THE DDM PROCESS NAME
002206  012767  000000G 175564                  MOV     #$TMLST,CURSLT  ; GET NEXT SLT FROM START OF LIST
                                        ;
                                        ; SEARCH FOR THE RIGHT CONTROLLER
                                        ;
002214  011505                          20$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
002216  001474                                  BEQ     65$             ; IF EQ, END OF TEMPLATES
002220  122765  000000G 000000G                 CMPB    #CS.CNT,C.STS(R5) ; IS THIS A CONTROLLER TEMPLATE ?
002226  001405                                  BEQ     30$             ; IF EQ, YES .. OKAY
002230  032767  000000G 000000G 25$:            BIT     #WC.CNT,$WILD   ; ELSE, IS IT ALL CONTROLLERS ?
002236  001766                                  BEQ     20$             ; IF EQ, NO .. LOOK FOR CONTROLLER
002240  000413                                  BR      40$             ; ELSE, MAYBE A UNIT TEMPLATE
002242  032767  000000G 000000G 30$:            BIT     #WC.CNT,$WILD   ; ALL CONTROLLERS ?
002250  001004                                  BNE     35$             ; IF NE, YES
002252  026567  000000G 000000G                 CMP     C.PRM(R5),$CNTNM ; IS THIS THE RIGHT CONTROLLER ?
002260  001355                                  BNE     20$             ; IF NE, NO .. KEEP LOOKING
002262  016567  000000G 000000G 35$:            MOV     C.PRM(R5),$CNTNM ; UPDATE CONTROLLER NUMBER
```

```
005112  016746  000000G                  MOV     $FLAGS,-(SP)     ; SAVE FLAGS
005116  005067  000000G                  CLR     $FLAGS           ; CLEAR LINE OPTIONS FLAG
005122  032765  000000G 000000G          BIT     #CS.UNI,C.STS(R5) ; MUX LINE ?
005130  001403                           BEQ     90$              ; IF EQ, NOPE
005132  052767  000000G 000000G          BIS     #FL.MUX,$FLAGS   ; ELSE FORMAT UNIT NUMBER
005140  004767  176402          90$:     CALL    FMTLN2           ; FORMAT THE LINE ID
005144  012667  000000G                  MOV     (SP)+,$FLAGS     ; RESTORE FLAGS
                                ;
                                ; DLM PVC - GET THE COST, HELLO TIMER
                                ;
005150  005067  174512                   CLR     PVCSW            ; ASSUME NOT DLM PVC
005154  026427  000000G 015355           CMP     P$VOWN(R4),#^RDLM ; CHECK FOR DLM
005162  001020                           BNE     100$             ; IF NE, NOT DLM
005164  010546                           MOV     R5,-(SP)         ; PUSH R5
005166  004767  177130                   CALL    GETSTA           ; GET STATION
005172  103413                           BCS     95$              ; NO STATION TEMPLATE ?
005174  012767  000707' 174464           MOV     #PVCMS6,PVCSW    ; ELSE PRINT STATION INFO
005202  012703  002022'                  MOV     #PVCBL6,R3       ; POINT TO ARG BLOCK
005206  116523  000000G                  MOVB    S$TCST(R5),(R3)+   ; SET COST,
005212  105023                           CLRB    (R3)+              ; ...
005214  016523  000000G                  MOV     S$THTM(R5),(R3)+   ; HELLO TIMER,
005220  005023                           CLR     (R3)+            ; NO MORE LISTEN TIMER.
005222  012605                  95$:     MOV     (SP)+,R5         ; RESTORE R5
                                ;
                                ; PRINT THE PVC CHARACTERISTICS
                                ;
005224  012703  001642'         100$:    MOV     #PVCTBL,R3       ; POINT TO CONTROL TABLE
005230  004767  000004                   CALL    PRINT            ; PRINT IT
005234  012603                  110$:    MOV     (SP)+,R3         ; RESTORE R3

005236  000207                  120$:    RETURN

```

```
                                        ; DEFINE NODE TEMPLATE
                                        ;
000022                                           .ASECT
          000000                                 .=0
000000                                           .BLKW   1               ; C.LNK
000002                                           .BLKW   1               ; C.STS
000004                                  C.NAM:   .BLKW   2               ; NODE NAME
000010                                  C.NID:   .BLKW   1               ; NODE ID
000012                                           .BLKB   32.             ; LENGTH AND STRING
000052                                  C.NUM:   .BLKW   1               ; NODE ADDRESS
000054                                  C.HOS:   .BLKW   1               ; HOST NODE ADDRESS
          000056                        C.LEN=.
000000                                           .PSECT

000000                                  TEMP:    .BLKW   1               ; ADDRESS OF REMOTE TEMPLATE
                                        ;
                                        ; DEFINE ROUTING TEMPLATE
                                        ;
000056                                           .ASECT
          000000                                 .=0
000000                                           .BLKW   1               ; C.LNK
000002                                           .BLKW   1               ; C.STS
000004                                  C.ADD:   .BLKW   1               ; MAXIMUM NODE ADDRESS
000006                                  C.CST:   .BLKW   1               ; MAXIMUM COST ALLOWED
000010                                  C.HOP:   .BLKW   1               ; MAXIMUM NUMBER OF HOPS ALLOWED
000012                                  C.RTM:   .BLKW   1               ; ROUTING TIMER
000014                                  C.NA:    .BLKW   1               ; NUMBER OF AREAS
000016                                  C.NBEA:  .BLKW   1               ; NUMBER BRO ENDNODE ADJACENCIES
000020                                  C.AHOP:  .BLKW   1               ; AREA HOPS
000022                                  C.ACST:  .BLKW   1               ; AREA COST
000024                                  C.L2TM:  .BLKW   1               ; LEVEL 2 TIMER
          000026                        C.LEN=.
000002                                           .PSECT

                                                 .NLIST  BEX
                                                 .ENABL  LC
          000002                                 EX$ERR = 2
000002       002       103       106    ERRXPT:  .ASCIZ  <EX$ERR>/CFE -- No XPT process macro/
000037       002       103       106    ERRECL:  .ASCIZ  <EX$ERR>/CFE -- No ECL process macro/
000074       002       103       106    ERRDEC:  .ASCIZ  <EX$ERR>/CFE -- No DEC$DF macro/
000124       002       103       106    ERRNOD:  .ASCIZ  <EX$ERR>/CFE -- No node definition/
000157       002       103       106    ERRROU:  .ASCIZ  <EX$ERR>/CFE -- No routing definition/
000215       012       105       170    HEADR:   .ASCIZ  <12>/Executor characteristics as of %Y %3Z/
000264       012       040       040    IDMSG:   .ASCIZ  <12>/  Identification = %VA/
000314       040       040       116    NAMMSG:  .ASCIZ  /  Name = %2R,/
000332       040       101       144    ADDMSG:  .ASCIZ  / Address = %M,/
000351       040       110       157    HSTMSG:  .ASCIZ  / Host = %M/
000364       040       101       144    ADDMS2:  .ASCIZ  / Address = %M.%M,/
000406       040       110       157    HSTMS2:  .ASCIZ  / Host = %M.%M/
000424       040       050       045    HSTNAM:  .ASCIZ  / (%2R),/
000434       040       040       115    HOSMSG:  .ASCIZ  /  Maximum links = %M,  Maximum node counters = %M/
000516       040       040       124    VFYMSG:  .ASCII  /  Type = %VA, Verification state = %VA/
000564       054       040       115    VFYCOM:  .ASCIZ  /, Maximum address = %M/
000613       040       040       123    ECLSEG:  .ASCIZ  /  Segment buffer size = %M/
000646       040       040       115    ROUMSG:  .ASCII  /  Maximum cost = %M, Maximum hops = %M%N/
000716       040       040       122             .ASCII  /  Routing timer = %M, Maximum broadcast nonrouters = %M%N/
001007       040       040       102             .ASCIZ  /  Broadcast routing timer = %M/
```

```
                                                                         L 15
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

HSTNAM     000424 R         #4-159     8-556
IDMSG      000264 R         #4-153     5-278
MSGBLK     001460 R         #4-183     5-270     5-283     5-297     5-307     5-315     5-339     5-349     5-382
                             5-389     5-397     5-402     5-412     6-455     7-477     8-511     8-553

NAMMSG     000314 R         #4-154     5-287
NF$ACC   = 100000            5-364
NF.ACC   = 000001            5-362
NF.END   = 000200            5-354
NF.LV2   = 000100            5-351     5-395
OFF        001432 R         #4-175     5-359
PRINT      002744 R          5-279     5-303     5-311     5-335     5-345     5-375     5-391     5-404     5-424
                            #6-453     8-515

PUTADR     002772 R          5-290    #7-475     8-533
PUTHST     003104 R          5-292    #8-532
P$STHI   = ******   GX       5-423
P$STLO   = ******   GX       5-417
RETMSG     001352 R         #4-173     5-344
ROUMSG     000646 R         #4-164     5-390
ROUTMP     001456 R         #4-182    *5-217     5-370     5-383     5-398
RO2MSG     001046 R         #4-167     5-403
SEGSIZ     003056 R          5-376    #8-509
SUBBA      001154 R         #4-169     5-419
SUBBA2     001201 R         #4-170     5-413    *5-422
TEMP       000000 R         #4-124
TIMMSG     001230 R         #4-171     5-302
VFYCOM     000564 R         #4-162    *5-369    *5-373     5-380
VFYMSG     000516 R         #4-161     5-374
$BUFF    = ******   GX       5-201     5-204     5-286     5-293     6-454     6-457
$CFERR   = ******   GX       5-430     5-432     5-434     5-436     5-438
$CNFIG   = ******   GX       5-351     5-354     5-395
$EDMSG   = ******   GX       5-203     5-288     6-456     7-496     8-557
$GTBUF   = ******   GX       5-202
$PRINT   = ******   GX       5-204     5-293     6-457
$PSIPT   = ******   GX       5-408
$RDBSZ   = ******   GX       8-514
$SHEXE     001500 RG        #5-199
$TMLST   = ******   GX       5-205     5-250     5-263     8-539
$$$      = 000023 R         #4-175     4-175     4-175    #4-176     4-176     4-176    #4-178     4-178     4-178
                            #4-179     4-179     4-179    #4-180     4-180     4-180

$$$L     = 000015           #4-175     4-175    #4-176     4-176    #4-178     4-178    #4-179     4-179    #4-180
                             4-180
```

```
                                                                         M 15
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26      PAGE 3
```

```
                                                .SBTTL  GETUNT - GET THE UNIT INFORMATION
                                        ;+
                                        ; *** - GETUNT - GET THE UNIT CHARACTERISTICS
                                        ;
                                        ; INPUT:
                                        ;       R5 = ADDRESS OF THE CURRENT UNIT DEFINITION
                                        ;
                                        ; OUTPUT:
                                        ;       UNTBLK = DUPLEX
                                        ;       UNTBLK+4 = THE LINE SPEED FOR THIS LINE
                                        ;       UNTBLK+10 = PROTOCOL (LAPB OR DDCMP)
                                        ;       OPTBLK = SECONDARY CSR AND/OR DEAD POLLING RATIO
                                        ;
                                        ;-

002260  012767  000013  175612  GETUNT: MOV     #OP.CNT!OP.UNI!OP.STA,DSPOPT ; DISPLAY CONTROLLER AND UNIT INFO
002266  012703  001446'                 MOV     #OPTBLK,R3      ; POINT AT OPTIONAL INFO BLOCK
002272  005023                          CLR     (R3)+           ; CLEAR OUT FIRST HALF OF BLOCK
002274  005013                          CLR     (R3)            ; ...
002276  005063  000002                  CLR     2(R3)           ; AND SECOND HALF OF BLOCK
002302  005063  000004                  CLR     4(R3)           ; ...
002306  112767  000040  176707          MOVB    #' ,CSR         ; ASSUME ONLY SECONDARY CSR IN OPTIONAL DATA
                                ;
                                ; DUPLEX
                                ;
002314  012767  000004  177110          MOV     #FULLN,UNTBLK   ; ASSUME FULL DUPLEX DEVICE
002322  012767  000730' 177104          MOV     #FULL,UNTBLK+2  ; ...
002330  032765  000001  000006          BIT     #CH.DUP,C.CWO(R5) ; FULL DUPLEX DEVICE ?
002336  001406                          BEQ     10$             ; IF EQ, YES .. OKAY
002340  012767  000004  177064          MOV     #HALFLN,UNTBLK  ; ELSE, HALF DUPLEX
002346  012767  000734' 177060          MOV     #HALF,UNTBLK+2  ; ...
                                ;
                                ; LINE SPEED
                                ;
002354  012767  000005  177054  10$:    MOV     #HISPDL,UNTBLK+4 ; ASSUME HIGH SPEED DEVICE
002362  012767  000754' 177050          MOV     #HISPD,UNTBLK+6 ; ...
002370  016501  000006                  MOV     C.CWO(R5),R1    ; GET CHARACTERISTICS WORD #0
002374  000301                          SWAB    R1              ; GET THE TRANS SPEED INDEX
002376  042701  177760                  BIC     #^C<17>,R1      ; ...
002402  020127  000017                  CMP     R1,#17          ; HIGH SPEED DEVICE ?
002406  001416                          BEQ     20$
002410  006301                          ASL     R1              ; CONVERT TO A WORD INDEX
002412  016101  000000'                 MOV     RATETB(R1),R1   ; GET SPEED FROM TABLE
002416  012700  001252'                 MOV     #SCRACH,R0      ; POINT AT SCRACH BUFFER
002422  010067  177012                  MOV     R0,UNTBLK+6     ; SAVE START ADDRESSOF BUFFER
002426  005002                          CLR     R2              ; ZERO SUPPRESSION
002430  004767  000000G                 CALL    $CBDMG          ; CONVERT TO ASCII
002434  166700  177000                  SUB     UNTBLK+6,R0     ; CALCULATE LENGTH OF BUFFER
002440  010067  176772                  MOV     R0,UNTBLK+4     ; SAVE IT IN ARGUMENT BUFFER
                                ;
                                ; PROTOCOL TYPE
                                ;
002444  012700  001264'         20$:    MOV     #HWP,R0         ; POINT TO DDM NAME BUFFER
002450  016701  000000C                 MOV     $DVNAM+L$NAM,R1 ; GET DEVICE NAME
002454  004767  000000G                 CALL    $C5TA           ; CONVERT IN CASE HARDWARE PROTOCOL
002460  016500  000010                  MOV     C.CW1(R5),R0    ; GET CHARACTERISTICS
002464  042700  177770                  BIC     #^C7,R0         ; ISOLATE PROTOCOL TYPE
```

```
001220  005201                  INC     R1              ; ELSE SET FLAG
                        ;
                        ; IS TASKNAME VALID FOR NUMNER OF COPIES ?
                        ;
001222  005702          30$:    TST     R2              ; PARSED A MULTICOPY NAME ?
001224  100403                  BMI     35$             ; IF MI, NO
001226  005701                  TST     R1              ; ELSE YES - MULTIPLE COPIES SPECIFIED ?
001230  001004                  BNE     50$             ; IF NE, YES - SUCCESS
001232  000402                  BR      40$             ; ELSE SET CARRY

001234  005701          35$:    TST     R1              ; MULTI COPIES SPECIFIED
001236  001401                  BEQ     50$             ; IF EQ, NO - SUCCESS

001240  000261          40$:    SEC                     ; OTHERWISE ERROR

001242  000207          50$:    RETURN

        000001                  .END
```

```
                                        .SBTTL  MACRO DEFINITIONS

                                        .MCALL  LLCTP$

 000000                                 LLCTP$  LIST            ; DEFINE LLC$DF TEMPLATE OFFSETS
                                ;
                                ; DEFINE LLC TEMPLATE
                                ;
 000000                                 .ASECT
          000000                        .=0
 000000                                 .BLKW   1               ; C.LNK
 000002                                 .BLKW   1               ; C.STS
 000004                         C.NAM:  .BLKW   1               ; LLC PROCESS NAME
 000006                         C.FLG:  .BLKW   1               ; LLC FLAGS WORD
 000010                         C.PRI:  .BLKB   1               ; LLC PROCESS PRIORITY
 000011                         C.LIN:  .BLKB   1               ; NUMBER OF LINES ASSIGNED TO THIS LLC
 000012                         C.DEV:  .BLKW   1               ; PSEUDO DEVICE NAME FOR THIS LLC
 000014                         C.NLT:  .BLKB   1               ; NUMBER OF LINE TABLE TO ALLOCATE
 000015                         C.EXT:  .BLKB   1               ; PROCESS EXTENSION IN BLOCKS
 000016                         C.CTM:  .BLKW   1               ; COUNTER TIMER
          000020                CL.LEN=.
 000000                                 .PSECT
          000012                C.NLTD = 12                     ; NLT FOR DDM/DLC PROCESSES
          000013                C.NCTD = 13                     ; NCT FOR DDM/DLC PROCESSES
                                ;****
                                ; LOCAL MACROS
                                ;****

                                        .MACRO  PRINT   TEXT
                                        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
                                        CALL    $PRINT          ; PRINT MESSAGE
                                        .ENDM   PRINT

                                        .MACRO  ERROR$  TEXT
                                ;
                                ; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
                                ; MESSAGE STRING.
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
                                        .ENDC
                                        CALL    $CFERR                  ; PRINT ERROR MESSAGE
                                        .ENDM   ERROR$

                                        .MACRO  EPRINT  TEXT
                                ;
                                ; PRINT TEXT ON ERROR LUN
                                ;
                                        .IF DIF <TEXT><R0>
                                        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
                                        .ENDC
                                        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
                                        .ENDM   EPRINT

                                        .NLIST  BEX
                                        .ENABL  LC
          000002                        EX$ERR = 2
```

```
001064  022767  000000G 000000G 75$:   CMP    #CO.KNC,$COMP    ; KNOWN CIRCUITS ?
001072  001407                         BEQ    80$              ; IF EQ, YES
001074  032767  000000G 000000G        BIT    #WC.CNT!WC.UNT!WC.TRI,$WILD ; WILDCARD/KNOWN OPERATION ?
001102  001003                         BNE    80$              ; IF NE, OK IF NOTHING FOUND
001104  005767  177214          77$:   TST    FOUND            ; ELSE WAS IT FOUND ?
001110  001413                         BEQ    101$             ; IF NE, NO - ERROR

001112  000241                  80$:   CLC                     ; RETURN SUCCESS
001114  000207                         RETURN


                                ;
                                ; ERROR CONDITIONS
                                ;
001116  012705  000145'         100$:  MOV    #ERRCHN,R5       ; CHANNEL NUMBER MISSING
001122  000426                         BR     120$

001124  012705  000106'         103$:  MOV    #ERRPAR,R5       ; DTE MISSING
001130  000423                         BR     120$

001132  012705  000000G         104$:  MOV    #ERRALL,R5       ; ALLOCATION FAILURE
001136  000420                         BR     120$

001140  012705  000000'         101$:  MOV    #ERRPVC,R5       ; PVC NOT FOUND
001144  000415                         BR     120$

001146  032767  000000G 000000G 110$:  BIT    #FL.CRE,$FLAGS   ; CREATED TEMPLATE?
001154  001407                         BEQ    115$             ; IF EQ, NO
001156  010002                         MOV    R0,R2            ; COPY TEMPLATE POINTER
001160  012701  000000G                MOV    #P$VLEN,R1       ; LENGTH TO RETURN
001164  012700  000000G                MOV    #$XXAVL,R0       ; POINT TO FREE LIST
001170  004767  000000G                CALL   $RLBF            ; RELEASE CORE BLOCK
001174  012705  000045'         115$:  MOV    #ERRDTE,R5       ; NO SUCH DTE

001200                          120$:  ERROR$ R5               ; PRINT ERROR MESSAGE
001206  000261                         SEC                     ; RETURN ERROR
001210  000207                         RETURN


```

```
                                                .SBTTL  $SEREM - UPDATE REMOTE DEFINITION
                                ;+
                                ; *** - $SEREM - UPDATE REMOTE DEFINITION
                                ;
                                ; INPUT:
                                ;       NONE.
                                ;
                                ; OUTPUT:
                                ;       THE REMOTE TEMPLATE IS MODIFIED
                                ;
                                ;-

000046                          $SEREM::
000046  012700  000000G                 MOV     #$TMLST,R0        ; GET START OF TEMPLATES
000052  011000                  5$:     MOV     (R0),R0           ; GET NEXT TEMPLATE
000054  001507                          BEQ     60$               ; NO DEC$DF MEANS PSI ONLY, DON'T ALLOW SET
000056  122760  000000G 000000G         CMPB    #CS.DEC,C.STS(R0) ; FOUND THE DEC$DF TEMPLATE YET?
000064  001372                          BNE     5$                ; IF NE, NO, KEEP LOOKING
                                ;                                 ; ELSE, FOUND IT, GO SEARCH FOR REMOTE
                                ; SEARCH FOR THE REMOTE TEMPLATE
                                ;
000066  011000                  10$:    MOV     (R0),R0           ; GET NEXT TEMPLATE
000070  001423                          BEQ     20$               ; IF EQ, NOT IN LIST .. MAKE NEW ENTRY
000072  122760  000000G 000000G         CMPB    #CS.REM,C.STS(R0) ; IS THIS A REMOTE TEMPLATE ?
000100  001372                          BNE     10$               ; IF NE, NO .. KEEP LOOKING
000102  026760  000000G 000010          CMP     $NODAD,C.ADD(R0)  ; IS THIS THE RIGHT TEMPLATE ?
000110  001366                          BNE     10$               ; IF NE, NO .. KEEP LOOKING
                                ;
                                ; FOUND THE REMOTE DEFINITION
                                ;
000112  032767  000000G 000000G         BIT     #RN.NAM,$OPRND    ; REMOTE NAME PARSED ?
000120  001463                          BEQ     50$               ; NOPE
000122  016760  000000G 000004          MOV     $NODNA,C.NAM(R0)  ; STORE THE NEW NODE NAME
000130  016760  000002G 000006          MOV     $NODNA+2,C.NAM+2(R0) ; ...
000136  000454                          BR      50$               ; AND LEAVE
                                ;
                                ; CREATE A NEW REMOTE TEMPLATE
                                ;
000140  012700  000000G         20$:    MOV     #$XXAVL,R0        ; POINT AT END OF TASK LISTHEAD
000144  012701  000012                  MOV     #C.LEN,R1         ; GET LENGTH OF A REMOTE TEMPLATE
000150  004767  000000G                 CALL    $RQBF             ; TRY TO ALLOCATE A TEMPLATE
000154  103005                          BCC     30$               ; IF CC, OKAY GOT ONE
000156                                  ERROR$  #ERRBUF           ; ALLOCATION FAILURE
000166  000440                          BR      50$               ; AND LEAVE
000170  012760  000000G 000000G 30$:    MOV     #CS.REM,C.STS(R0) ; INDICATE THAT THIS IS A REMOTE TEMPLATE
000176  016760  000000G 000004          MOV     $NODNA,C.NAM(R0)  ; STORE THE REMOTE NAME
000204  016760  000002G 000006          MOV     $NODNA+2,C.NAM+2(R0) ; ...
000212  032767  176000  000000G         BIT     #^C1023.,$NODAD   ; AREA SPECIFIED ?
000220  001006                          BNE     35$               ; IF NE, YES - USE IT
000222  016746  000000G                 MOV     $NTADD,-(SP)      ; ELSE PUSH EXEC ADDRESS
000226  042716  001777                  BIC     #1023.,(SP)       ; ISOLATE AREA BITS
000232  052667  000000G                 BIS     (SP)+,$NODAD      ; DO WHAT HE MEANS, NOT WHAT HE SAYS
000236  016760  000000G 000010  35$:    MOV     $NODAD,C.ADD(R0)  ; STORE THE NODE ADDRESS
000244  004767  000026                  CALL    FINREM            ; FIND WHERE REMOTE SHOULD GO
000250  004767  000000G                 CALL    $QINSN            ; INSERT ENTRY IN LIST
000254  062767  000000G 000000G         ADD     #R.LENX,$BYTMN    ; UPDATE POOL BYTE EXTENSION MINIMUM
000262  062767  000000G 000             ADD     #R.LENX,$BYTXT    ; AND CURRENT POOL BYTE VALUE
```

M 5

```
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37       PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE             REFERENCES

SF.DEV  = 000020          #4-136     7-295
SF.DPA  = 000100          #4-138     7-297
SF.DPC  = 000200          #4-139     7-298
SF.HAD  = 000004          #4-134     7-300
SF.HST  = 000002          #4-133     7-294
SF.PH3  = 000400          #4-140     7-296
SF.PSS  = 000040          #4-137     7-301
S$ADD     000006          #4-119     12-511    *12-527
S$CIR     000022          #4-122     7-299
S$DEV     000032          #4-123     7-295
S$DPA     000050          #4-125     7-297
S$DPC     000054          #4-126     7-298
S$FLG     000004          #4-118     7-296     *8-327     *12-528
S$HAD     000012          #4-121     7-300
S$HST     000010          #4-120     7-294
S$LEN   = 000056          #4-127     12-523
S$PSS     000034          #4-124     7-301
S.LENX  = ******   GX      12-532    12-533
UPDDLL    000332 R         5-236    #7-274
$BYTMN  = ******   GX     *5-234    *8-375     *9-412     *12-532
$BYTXT  = ******   GX     *5-235    *8-376     *9-413     *12-533
$CFERR  = ******   GX      5-221     11-474     12-553
$FILHD  = ******   GX      9-400     10-435     11-470    *11-471
$NODAD  = ******   GX      5-204     5-226     *5-230      5-231
$NODNA  = ******   GX      5-211     5-212      5-224      5-225
$NTADD  = ******   GX      5-228
$OPRND  = ******   GX      5-209     7-274      7-285
$QINSN  = ******   GX      5-233     12-531     12-546
$RLBF   = ******   GX      9-410
$RNDCP  = ******   GX      7-298
$RNDIA  = ******   GX      7-305
$RNDMP  = ******   GX      7-306
$RNDPA  = ******   GX      7-297
$RNHAR  = ******   GX      7-300
$RNHOS  = ******   GX      7-294
$RNLOD  = ******   GX      7-304
$RNSCI  = ******   GX      7-299
$RNSDV  = ******   GX      7-295
$RNSLO  = ******   GX      7-302
$RNSPA  = ******   GX      7-301
$RNTLO  = ******   GX      7-303
$RQBF   = ******   GX      5-219     11-467     12-524     12-537
$SAVAL  = ******   GX      9-396     10-434
$SAVVR  = ******   GX      11-464
$SEREM    000046 RG       #5-191
$SWICH  = ******   GX      7-296
$TMLST  = ******   GX      5-192     6-253      12-506
$XXAVL  = ******   GX      5-217     9-407      11-465     12-522     12-535
.CONER  = ******   GX      9-416     12-551
.EXIT   = ******   GX      11-475    12-554
```

N 5

```
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37       PAGE 3
```

Table of contents

```
    4-  118    $SESLT - SET SLT PARAMETERS
    5-  184    SETLIN - SET LINE STATE
    6-  233    SETCIR  - SET CIRCUIT STATE
    7-  270    SETPTP - SET POINT TO POINT LINE/CIRCUIT STATE
    8-  332    SETSTA - SET STATION PARAMETERS
    9-  381    FNDACT - FIND TRIBUTARY ACTIVITY
   10-  416    ACTERR - MULTIPOINT TRIBS ACTIVE ERROR
   11-  446    CHKSTA - CHECK LINE/CIRCUIT STATES
```

```
                                                .SBTTL  CHKSTA - CHECK LINE/CIRCUIT STATES
                                        ;+
                                        ; **-CHKSTA-CHECK LINE/CIRCUIT STATES
                                        ;
                                        ; THIS ROUTINE CHECKS FOR MULTIPLE ACTIVE LINES
                                        ; ON ENDNODES.
                                        ;
                                        ; ALL REGISTERS ARE PRESERVED.
                                        ;-

001334                                  CHKSTA:
001334  004567  000000G                         JSR     R5,$SAVRG       ; SAVE REGISTERS
001340  016746  000000G                         MOV     $FLAGS,-(SP)    ; SAVE PARSE FLAGS
001344  016746  000000G                         MOV     $STAN,-(SP)     ; ..STATION NUMBER
001350  016746  000000G                         MOV     $WILD,-(SP)     ; ..WILDCARDS

001354  032767  000000C 000000G                 BIT     #OC.STA!OL.STA,$OPRND ; STATE CHANGE REQUESTED ?
001362  001473                                  BEQ     70$             ; NO
001364  032767  000000G 000000G                 BIT     #NF.END,$CNFIG  ; IS THIS AN END NODE ?
001372  001467                                  BEQ     70$             ; NO
001374  032767  000000G 000000G                 BIT     #SW.ENA,$SWICH  ; REQUESTED STATE = ON ?
001402  001463                                  BEQ     70$             ; NO
001404  005767  000000G                         TST     $WILD           ; WILDCARDS SPECIFIED ?
001410  001052                                  BNE     40$             ; IF NE, YES - ERROR

001412  016705  000000G                         MOV     $TMLST,R5       ; POINT TO FIRST TEMPLATE
001416  005004                                  CLR     R4              ; NO SLT POINTER YET
001420  005003                                  CLR     R3              ; OR TRIB POINTER, EITHER
001422  004767  000000G                         CALL    .FNSLT          ; LOCATE TARGET SLT
001426  103451                                  BCS     70$             ; SHOULDN'T EVER HAPPEN, BUT...
001430  010504                                  MOV     R5,R4           ; SAVE TARGET SLT ADDRESS
001432  032767  000000G 000000G                 BIT     #LF.MTP,$FLAGS  ; IS IT MULTIPOINT ?
001440  001403                                  BEQ     10$             ; IF EQ, NO
001442  004767  000000G                         CALL    $FNSTA          ; ELSE LOCATE STATION
001446  010503                                  MOV     R5,R3           ; SAVE STATION ADDRESS

001450  016705  000000G                 10$:    MOV     $TMLST,R5       ; POINT TO FIRST TEMPLATE
001454  012767  000000G 000000G                 MOV     #WC.CIR,$WILD   ; SCAN WILDCARD CIRCUITS

001462  004767  000000G                 20$:    CALL    .FNSLT          ; LOOK FOR NEXT SLT
001466  103431                                  BCS     70$             ; IF CS, END OF THE LINE AND NO ERROR
001470  032767  000000G 000000G                 BIT     #FL.MTP,$FLAGS  ; MULTIPOINT CIRCUIT ?
001476  001007                                  BNE     30$             ; IF NE, YES
001500  020504                                  CMP     R5,R4           ; IS THIS THE TARGET CIRCUIT ?
001502  001767                                  BEQ     20$             ; IF EQ, IT IS - DON'T CARE
001504  032765  000000G 000000G                 BIT     #LF.ENA,S$LFLG(R5) ; ELSE, IS IT ALREADY ON ?
001512  001011                                  BNE     40$             ; IF NE, YES - ERROR
001514  000762                                  BR      20$             ; CONTINUE

001516  004767  000000G                 30$:    CALL    $FNSTA          ; LOCATE STATION TEMPLATE
001522  020503                                  CMP     R5,R3           ; IS THIS THE TARGET CIRCUIT ?
001524  001756                                  BEQ     20$             ; YES - DON'T CARE
001526  132765  000000G 000000G                 BITB    #SF.ENA,S$TFLG(R5) ; STATION ALREADY ACTIVE ?
001534  001752                                  BEQ     20$             ; NO - ONWARD

001536                                  40$:    ERROR$  #ERRSTA         ; ERROR - WILDCARDS NOT ALLOWED
001546  000261                                  SEC
```

```
SETSVC - SET SVC PARAMETERS     MACRO V05.03b Saturday 29-Jun-85 05:24  Page 5-3
Symbol table

CO.KNC= ****** GX      OC.USG= ****** GX      S$DTE    000020         S$VFLG= ****** GX      $FLAGS= ****** GX
CS.SVC= ****** GX      SCR      000110R       S$FLG    000010         S$VNAM= ****** GX      $FNSLT= ****** GX
C.STS = ****** GX      SF$BLK= 000001         S$LEN    000032         S$VPSZ= ****** GX      $FNSTA= ****** GX
ERRCIS= ****** GX      SF$INC= 000002         S$LFT    000012         S$VRCL= ****** GX      $MBL   = ****** GX
ERRSVC  000000R        SF.ENA= ****** GX      S$LNK    000000         S$VTIM= ****** GX      $MRC   = ****** GX
ERRUSG  000045R        SV$DTE= ****** GX      S$OWNR   000030         S$VWSZ= ****** GX      $MWI   = ****** GX
EX$ERR= 000002         SV$PSZ= ****** GX      S$PKSZ   000004         S$WND   000006         $OPRND= ****** GX
FL.DLM= ****** GX      SV$WSZ= ****** GX      S$RCL    000013         WC.CNT= ****** GX      $RTM   = ****** GX
OC.MBL= ****** GX      SW.ENA= ****** GX      S$STA    000003         WC.TRI= ****** GX      $SESVC  000112RG
OC.MRC= ****** GX      SW.INC= ****** GX      S$TFLG= ****** GX       $CFERR= ****** GX      $STAN = ****** GX
OC.MWI= ****** GX      SW.OUT= ****** GX      S$TIM    000014         $CNTNM= ****** GX      $SWICH= ****** GX
OC.NUM= ****** GX      S$ADSZ  000007         S$TMR    000016         $COMP  = ****** GX     $TMLST= ****** GX
OC.RTM= ****** GX      S$CTL   000002         S$VDTE= ****** GX       $DTE   = ****** GX     $WILD = ****** GX
OC.STA= ****** GX

. ABS.  000032    000   (RW,I,GBL,ABS,OVR)
        000636    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9125 Words  ( 36 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.48
SY:SETSVC.V2,[132,134]SETSVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETSVC
```

```
SETSVC     CREATED BY  MACRO  ON 29-JUN-85 AT 05:24     PAGE 1
```

```
                                                                        M  9
SETUNT      CREATED BY  MACRO  ON 12-AUG-85 AT 15:02        PAGE 1
SYMBOL CROSS REFERENCE                                      CREF    04.00

SYMBOL   VALUE             REFERENCES

CS.CNT   = ******   GX      5-178       5-194
CS.DDM   = ******   GX      5-167
CS.MOD   = ******   GX      5-207       6-272
CS.UNT   = ******   GX      5-190
C.CSR      000012          #4-137      *5-230
C.CST      000014          #4-138      *5-210
C.CWO      000006          #4-135      *5-214      *5-217      *7-322      *7-323      *7-327
C.CW1      000010          #4-136
C.DPR      000016          #4-139      *5-234
C.LEN    = 000024          #4-142
C.PPM    = ******   GX      5-169       5-171       5-182
C.STS    = ******   GX      5-167       5-178       5-190       5-194      *5-207      *5-211      *5-231      *5-235      *6-272
C.UNT      000004          #4-134       5-202       5-224
DDM        000000 R        #4-97       *5-169
END        000132 R        #4-123       4-125
ERRBAU     000002 R        #4-102       7-318
ERRSEC     000034 R        #4-103       5-227
EX$ERR   = 000002          #4-100       4-102
EX$WAR   = 000000          #4-101       4-103
FL.KMX   = ******   GX      5-222
FL.LMC   - ******   GX      5-220
FL.MUX   = ******   GX      5-198
OC.APR   = ******   GX      6-254       6-282
OC.CST   = ******   GX      5-208       6-254       6-276
OC.TRI   = ******   GX      6-254       6-288
OL.BAU   = ******   GX      5-236
OL.DPR   = ******   GX      5-232
OL.PLX   = ******   GX      5-212
OL.SEC   = ******   GX      5-218       6-256
RATETB     000076 R        #4-108       4-125       7-313
SETSTA     000564 R         5-159      #6-253
SPEED      000714 R         5-238      #7-309
SW.FUL   = ******   GX      5-215
TBLEN    = 000034          #4-125       7-309       7-316
UN$CST   = ******   GX      5-211
UN$DPR   = ******   GX      5-235
UN$SCS   = ******   GX      5-231
WC.CNT   = ******   GX      5-180       5-192
WC.UNT   = ******   GX      5-200
$APR     = ******   GX      6-284
$BAUD    = ******   GX      7-310       7-313
$CFERR   = ******   GX      5-227       7-318
$CNTNM   = ******   GX      5-182
$COST    = ******   GX      5-210       6-278
$DPR     = ******   GX      5-234
$FLAGS   = ******   GX     *5-170      *5-171       5-198       5-220       5-222
$FNPRO   = ******   GX      5-164
$FNSTA   = ******   GX      6-269
$OPRND   = ******   GX      5-208       5-212       5-218       5-232       5-236       6-254       6-256       6-276       6-282
                            6-288
$SCSR    = ******   GX      5-230
$SEUNT     000134 RG       #5-158
```

```
                                                                        N  9
SETUNT      CREATED BY  MACRO  ON 12-AUG-85 AT 15:02        PAGE 2
```

```
                .TITLE  SETX3P - SET X.25 LEVEL 3 PARAMETERS
                .IDENT  /V05.00/
                .ENABL  LC

;
; COPYRIGHT (C) 1981, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; CFE - SET X.25 LEVEL 3 PARAMETERS
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  26-OCT-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                .SBTTL  $SHOBJ - SHOW OBJECT CHARACTERISTICS
                                        ;+
                                        ; *** - $SHOBJ - SHOW OBJECT CHARACTERISTICS
                                        ;
                                        ; INPUT:
                                        ;       $TNUM = THE TASK NUMBER TO DISPLAY (-1 IMPLIES KNOWN OBJECTS)
                                        ;       $WILD = WILD CARD SPECIFICATIONS
                                        ;
                                        ; OUTPUT:
                                        ;       THE OBJECT CHARACTERISTICS ARE DISPLAYED.
                                        ;       R0-R5 = DESTROYED
                                        ;
                                        ;-

000450                                  $SHOBJ::
000450  005000                                  CLR     R0              ; SUCCESS INDICATOR
000452  005067  177660                          CLR     HFLAG           ; CLEAR PRINT HEADER FLAG
000456  012705  000000G                         MOV     #$TMLST,R5      ; GET START OF TEMPLATE LIST
                                        ;
                                        ; SEARCH FOR THE OBJECT TEMPLATE
                                        ;
000462  011505                          10$:    MOV     (R5),R5         ; GET NEXT TEMPLATE
000464  001564                                  BEQ     60$             ; IF EQ, END OF TEMPLATES
000466  122765  000000G 000000G                 CMPB    #CS.OBJ,C.STS(R5) ; IS THIS AN OBJECT TEMPLATE ?
000474  001372                                  BNE     10$             ; IF NE, NO .. KEEP LOOKING
000476  032767  000000G 000000G                 BIT     #WC.OBJ,$WILD   ; DO IT FOR ALL OBJECTS ?
000504  001004                                  BNE     20$             ; IF NE, YES
000506  126765  000000G 000004                  CMPB    $TNUM,C.TYP(R5) ; IS THIS THE RIGHT TEMPLATE ?
000514  001362                                  BNE     10$             ; IF NE, NO
                                        ;
                                        ; DISPLAY THE OBJECT CHARACTERISICS
                                        ;
000516  005767  177614                  20$:    TST     HFLAG           ; PRINT HEADER ?
000522  001024                                  BNE     27$             ; IF NE, NO
000524  005267  177606                          INC     HFLAG           ; ELSE, INDICATE HEADER ALREADY PRINTED
000530  012701  000044'                         MOV     #HEADR,R1       ; ASSUME FOR ONLY 1 OBJECT
000534  032767  000000G 000000G                 BIT     #WC.OBJ,$WILD   ; IS IT FOR ALL OBJECTS ?
000542  001402                                  BEQ     25$             ; IF EQ, NO .. OKAY
000544  012701  000111'                         MOV     #HEADR1,R1      ; ELSE, DIFFERENT HEADER
000550  012700  000000G                 25$:    MOV     #$BUFF,R0       ; POINT TO OUTPUT BUFFER
000554  012702  000000G                         MOV     #$GTBUF,R2      ; POINT AT ARGUMENT BLOCK
000560  004767  000000G                         CALL    $EDMSG          ; FORMAT THE HEADER
000564                                          PRINT   #$BUFF          ; PRINT MESSAGE
000574  005067  177540                  27$:    CLR     MS1BLK          ; CLEAR GARBAGE IN HIGH BYTE
000600  116567  000004  177532                  MOVB    C.TYP(R5),MS1BLK ; STORE THE OBJECT TYPE
000606  016567  000006  177526                  MOV     C.NAM(R5),MS2BLK ; STORE THE OBJECT NAME
000614  016567  000010  177522                  MOV     C.NAM+2(R5),MS2BLK+2 ; 2ND HALF OF NAME
000622  116501  000005                          MOVB    C.FLG(R5),R1    ; GET THE VERIFICATION LEVEL
000626  042701  177770                          BIC     #^C<7>,R1       ; CLEAN OUT GARBAGE BITS
000632  006301                                  ASL     R1              ; CONVERT TO A DOUBLE WORD INDEX
000634  006301                                  ASL     R1              ; ...
000636  016167  000410' 177502                  MOV     VTBL(R1),MS2BLK+4 ; GET LENGTH OF VERIFICATION MESSAGE
000644  016167  000412' 177476                  MOV     VTBL+2(R1),MS2BLK+6 ; AND ADDRESS OF MESSAGE
000652  012767  000005  177472                  MOV     #LOGINL,MS3BLK  ; ASSUME LOGIN ACCESS
000660  012767  000304' 177466                  MOV     #LOGIN,MS3BLK+2 ; ...
000666  132765  000100  000005                  BITB    #OF.RLU,C.FLG(R5) ; IS THIS LOGIN ACCESS ?
000674  001006                                  BNE     30$             ; IF NE, YES
```

```
                                          ;
                                          ; LOOK FOR THE UNIT TEMPLATE
                                          ;
002270  010500                            40$:    MOV     R5,R0             ; SAVE CURRENT TEMPLATE
002272  011505                                    MOV     (R5),R5           ; GET NEXT TEMPLATE
002274  001441                                    BEQ     60$               ; IF EQ, END OF TEMPLATES
002276  122765  000000G 000000G           45$:    CMPB    #CS.UNT,C.STS(R5) ; IS THIS A UNIT TEMPLATE ?
002304  001411                                    BEQ     50$               ; IF EQ, YES
002306  032767  000000G 000000G                   BIT     #WC.CNT,$WILD     ; MORE CONTROLLERS ?
002314  001435                                    BEQ     65$               ; IF EQ, NO .. DISPLAYED ALREADY
002316  122765  000000G 000000G                   CMPB    #CS.CNT,C.STS(R5) ; ELSE, IS IT A CONTROLLER TEMPLATE ?
002324  001756                                    BEQ     35$               ; IF EQ, YES
002326  000677                                    BR      10$               ; ELSE, GET NEXT DEVICE
002330  032767  000000G 000000G           50$:    BIT     #FL.MUX,$FLAGS    ; IS THIS A MUX DEVICE ?
002336  001410                                    BEQ     55$               ; IF EQ, NO .. MUST BE RIGHT UNIT
002340  032767  000000G 000000G                   BIT     #WC.UNT,$WILD     ; ALL UNITS ?
002346  001004                                    BNE     55$               ; IF NE, YES
002350  026567  000000G 000000G                   CMP     C.PRM(R5),$UNTNM  ; ELSE, IS THIS THE RIGHT UNIT ?
002356  001344                                    BNE     40$               ; IF NE, NO .. KEEP LOOKING
002360  016567  000000G 000000G           55$:    MOV     C.PRM(R5),$UNTNM  ; STORE THE UNIT NUMBER
002366  004767  000332                            CALL    GETTCI            ; GET TRANSPORT CIRCUIT INFORMATION
002372  004767  000106                            CALL    DISPLY            ; ELSE DISPLAY CIRCUIT CHARACTERISTICS
002376  000734                                    BR      40$               ; AND LOOK FOR NEXT UNIT
002400  022767  000000G 000000G           60$:    CMP     #CO.KNC,$COMP     ; KNOWN CIRCUITS ?
002406  001035                                    BNE     90$               ; IF NE, NO
                                          ;
                                          ; IF PSI IN SYSTEM, DISPLAY VIRTUAL CIRCUITS
                                          ;
002410  012705  000000G                   65$:    MOV     #$PSIPT,R5        ; POINT TO PSI TEMPLATES
002414  011505                            70$:    MOV     (R5),R5           ; POINT TO NEXT TEMPLATE
002416  001427                                    BEQ     85$               ; IF EQ, NO MORE TEMPLATES
002420  122765  000000G 000000G                   CMPB    #CS.X3P,C.STS(R5) ; IS IT X3P$DF ?
002426  001001                                    BNE     75$               ; IF NE, NO
002430  010503                                    MOV     R5,R3             ; ELSE SAVE IT
002432  122765  000000G 000000G           75$:    CMPB    #CS.DTE,C.STS(R5) ; IS THIS A DTE$DF TEMPLATE
002440  001365                                    BNE     70$               ; IF NE, NOT A DTE$DF TEMPLATE
002442  010504                                    MOV     R5,R4             ; COPY POINTER
002444  011404                            80$:    MOV     (R4),R4           ; NEXT TEMPLATE
002446  001413                                    BEQ     85$               ; IF EQ, END OF LIST
002450  122764  000000G 000000G                   CMPB    #CS.DTE,C.STS(R4) ; CHECK FOR DTE$DF
002456  001756                                    BEQ     70$               ; IF EQ, PROCESS NEXT DTE$DF SEQUENCE
002460  122764  000000G 000000G                   CMPB    #CS.PVC,C.STS(R4) ; CHECK FOR PERMANENT VIRTUAL CIRCUIT
002466  001366                                    BNE     80$               ; IF NE, NEXT TEMPLATE
002470  004767  001664                            CALL    SHOPVC            ; ELSE SHOW CHARACTERISTICS
002474  000763                                    BR      80$               ; CONTINUE
002476  004767  001170                    85$:    CALL    SHOSVC            ; DISPLAY ANY SVCS
002502  000207                            90$:    RETURN
```

```
                                       .SBTTL  PRINT - PRINT FORMATTED DATA
                                ;+
                                ; ** PRINT - PRINT FORMATTED DATA
                                ;
                                ; INPUTS:        R3 = ADDRESS OF FORMATTING TABLE
                                ;
                                ; OUTPUT:        INFORMATION PRINTED.
                                ;
                                ; REGISTERS:     R0, R1, R2, R3 CLOBBERED.
                                ;-

005240                          PRINT:
005240  012700  000000G         10$:   MOV     #$BUFF,R0       ; POINT TO BUFFER
005244  012301                         MOV     (R3)+,R1        ; POINT TO FORMAT STRING
005246  001410                         BEQ     20$             ; IF EQ, END OF TABLE
005250  012302                         MOV     (R3)+,R2        ; POINT TO ARG BLOCK
005252  004767  000000G                CALL    $EDMSG          ; FORMAT
005256                                 PRINT   #$BUFF          ; PRINT
0052[illegible]  000764                BR      10$             ; CONTINUE

005270  000207                  20$:   RETURN

        000001                         .END
```

```
    167 001046     040     040     115  RO2MSG: .ASCII  /  Maximum area = %M, Area maximum hops = %M%N/
    168 001123     040     040     101          .ASCIZ  /  Area maximum cost = %M/
    169 001154     040     040     114  SUBBA:  .ASCII  /  Low subaddress = %M/
    170 001201     054     040     110  SUBBA2: .ASCIZ  /, High subaddress = %M/
    171 001230     040     040     111  TIMMSG: .ASCIZ  /  Incoming timer = %M, Outgoing timer = %M/
    172 001303     040     040     104  DELMSG: .ASCIZ  /  Delay factor = %M, Delay weight = %M/
    173 001352     040     040     111  RETMSG: .ASCIZ  /  Inactivity timer = %M, Retransmit factor = %M/
    174                                         .EVEN
    175 001432                          OFF:    STRING  <Off>
    176 001436                                  STRING  <On>
    177
    178 001442                          ATYP:   STRING  <Area>
    179 001446                                  STRING  <Routing IV>
    180 001452                                  STRING  <Nonrouting IV>
    181
    182 001456                          ROUTMP: .BLKW   1               ; ROUTING TEMPLATE ADDRESS
    183 001460                          MSGBLK: .BLKW   10              ; MESSAGE ARGUMENT BLOCK
    184 001500                                  .PSECT
    185
```

```
SHOEXE      CREATED BY  MACRO  ON 29-JUN-85 AT 05:26     PAGE 3

MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME        REFERENCES

DECTP$            #4-59      4-61
EPRINT            #4-85
ERROR$            #4-74      5-430      5-432      5-434      5-436      5-438
MSGDF$            #4-59      4-63
NSFDF$            #4-59      4-62
PRINT             #4-69      5-204      5-293      6-457
STRING            #4-95      4-175      4-176      4-178      4-179      4-180
```

```
**FILE**ID**SHOLIN
```

```
002470  006300                           ASL     R0                     ; MAKE IT A WORD INDEX
002472  006300                           ASL     R0                     ;
002474  016067  001270' 176740           MOV     PROTBL(R0),UNTBLK+10 ; GET LENGTH
002502  016067  001272' 176734           MOV     PROTBL+2(R0),UNTBLK+12 ; STRING ADDRESS
002510  001023                           BNE     40$                    ; IF NE, OK
002512  005000                           CLR     R0                     ; ELSE ASSUME DDCMP POINT
002514  032765  000010  000010           BIT     #CH.TYP,C.CW1(R5) ; CHECK FOR MULTIPOINT
002522  001406                           BEQ     30$                    ; IF EQ, NO - DDCMP POINT
002524  005200                           INC     R0                     ; ELSE ASSUME TRIB
002526  032765  000020  000010           BIT     #CH.MDT,C.CW1(R5) ; TRIBUTARY ?
002534  001001                           BNE     30$                    ; IF NE, PROTOCOL = DDCMP TRIBUTARY
002536  005200                           INC     R0                     ; ELSE PROTOCOL = DDCMP CONTROL
002540  006300                   30$:    ASL     R0                     ; MAKE IT A WORD INDEX
002542  006300                           ASL     R0                     ; DOUBLE WORD
002544  016067  001330' 176670           MOV     DDCTBL(R0),UNTBLK+10 ; STRING LENGTH
002552  016067  001332' 176664           MOV     DDCTBL+2(R0),UNTBLK+12 ; STRING ADDRESS
                                 ;
                                 ; DEAD POLLING RATIO
                                 ;
002560  016501  000016           40$:    MOV     C.DPR(R5),R1           ; GET THE DEAD POLLING RATIO
002564  001421                           BEQ     50$                    ; IF EQ, NONE
002566  012713  001164'                  MOV     #DPR,(R3)              ; SAVE ADDRESS OF PRINT BUFFER
002572  012700  001211'                  MOV     #DPRVAL,R0             ; GET ADDRESS OF BUFFER
002576  005002                           CLR     R2                     ; ZERO SUPPRESSION
002600  004767  000000G                  CALL    $CBDMG                 ; CONVERT TO ASCII
002604  162300                           SUB     (R3)+,R0               ; CALCULATE STRING LENGTH
002606  010063  177774                   MOV     R0,-4(R3)              ; SAVE LENGTH IN ARGUMENT BLOCK
002612  005023                           CLR     (R3)+                  ; SKIP TO SECONDARY CSR FIELD
002614  112767  000054  176401           MOVB    #<',>,CSR              ; ASSUME SECONDARY CSR IS PRESENT
002622  052767  000004  175250           BIS     #OP.OPT,DSPOPT         ; SELECT LINE FOR DISPLAY
                                 ;
                                 ; SECONDARY CSR
                                 ;
002630  016501  000012           50$:    MOV     C.CSR(R5),R1           ; GET SECONDARY CSR
002634  001414                           BEQ     70$                    ; IF EQ NONE
002636  012713  001223'                  MOV     #CSR,(R3)              ; SAVE ADDRESS IN ARGUMENT BLOCK
002642  012700  001240'                  MOV     #CSRVAL,R0             ; POINT AT OUTPUT BUFFER
002646  005002                           CLR     R2                     ; ZERO SUPPRESSION
002650  004767  000000G                  CALL    $CBOMG                 ; CONVERT TO ASCII
002654  161300                           SUB     (R3),R0                ; GET LENGTH OF STRING
002656  010043                           MOV     R0,-(R3)               ; STORE IT IN ARGUMENT BLOCK
002660  052767  000004  175212           BIS     #OP.OPT,DSPOPT         ; SELECT LINE FOR DISPLAY

002666  000207                   70$:    RETURN
```

```
                                                                                N 1
SETOBJ - SET OBJECT OPTIONS     MACRO V05.03b Saturday 29-Jun-85 05:22  Page 7-2
Symbol table

CHKMUL  001056R         ERRNAM  000042R         OO.USR= ****** GX       SW.RLU= ****** GX       $RQBF = ****** GX
CS.OBJ= ****** GX       ERRONA  000115R         OO.VFY= ****** GX       SW.SMC= ****** GX       $SAVVR= ****** GX
C.CPY   000012          EX$ERR= 000002          O.FLG   000003          $BYTMN= ****** GX       $SEOBJ  000252RG
C.FLG   000005          FINOBJ  001022R         O.LEN   000012          $BYTXT= ****** GX       $SWICH= ****** GX
C.LEN = 000013          OBJNAM  000244R         O.LENX= ****** GX       $CFERR= ****** GX       $TMLST= ******
C.NAM   000006          OF.PRO= 000040          O.LNK   000000          $COPES= ****** GX       $TNAM = ****** GX
C.STS = ****** GX       OF.RLU= 000100          O.MXC   000004          $C5TA = ****** GX       $TNUM = ****** GX
C.TYP   000004          OF.SMC= 000200          O.NAM   000006          $OPRND= ****** GX       $VLVL = ****** GX
ERRBUF  000000R         OO.COP= ****** GX       O.TYP   000002          $QINSN= ****** GX       $XXAVL= ****** GX
ERRMUL  000201R         OO.NAM= ****** GX

. ABS.  000013    000   (RW,I,GBL,ABS,OVR)
        001244    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10380 Words  ( 41 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.40
SY:SETOBJ.V2,[132,134]SETOBJ/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]SETOBJ
```

```
     91 000000     002     103     106  ERRNSP: .ASCIZ  <EX$ERR>/LFE -- XPT not allowed to load line tables in process space/
     92                                         .LIST   BEX
     93                                         .EVEN
```

```
                                                   .SBTTL  SETDTE - SET NEW DTE ADDRESS
                                          ;+
                                          ; ** SETDTE - SET NEW DTE ADDRESS
                                          ;
                                          ; INPUT           R0 = ADDRESS OF PVC DESCRIPTOR
                                          ;                 R1 = PREVIOUS TEMPLATE ADDRESS FOR UNLINK
                                          ;                    = 0 IF TEMPLATE CREATED
                                          ;                 $FLAGS  SW.CRE SET IF TEMPLATE WAS CREATED
                                          ;                 $DTE = PARSED DTE STRING
                                          ;
                                          ; OUTPUT          CARRY CLEAR: TEMPLATE LINKED AFTER NEW DTE
                                          ;                              R0 = UNLINK ADDRESS OR ZERO
                                          ;                 CARRY SET: FAILURE
                                          ;
                                          ; REGISTERS       USES R2-R5
                                          ;-

 001212                                   SETDTE:
 001212  012705  000000G                          MOV     #$TMLST,R5        ; POINT TO TEMPLATES
 001216  011505                           10$:    MOV     (R5),R5           ; NEXT TEMPLATE
 001220  001434                                   BEQ     101$              ; IF EQ, NOT FOUND
 001222  122765  000000G 000000G                  CMPB    #CS.DTE,C.STS(R5); IS IT A DTE TEMPLATE?
 001230  001372                                   BNE     10$               ; BR IF NO
 001232  012702  000000G                          MOV     #$DTE,R2          ; POINT TO PARSED DTE ADDRESS
 001236  010503                                   MOV     R5,R3             ; POINT TO ADDESS STRING
 001240  062703  000000G                          ADD     #D$TADR,R3        ; ...
 001244  012704  000005                           MOV     #5,R4             ; STRING LENGTH
 001250  022223                           20$:    CMP     (R2)+,(R3)+       ; COMPARE STRINGS
 001252  001361                                   BNE     10$               ; BR IF WRONG DTE
 001254  005304                                   DEC     R4                ; REDUCE COUNT
 001256  003374                                   BGT     20$               ; UNTIL DONE
                                          ;
                                          ; DTE FOUND - IF PVC EXISTS, UNLINK FROM OLD DTE
                                          ;
 001260  005701                                   TST     R1                ; UNLINK
 001262  001404                                   BEQ     25$               ; IF EQ, NO
 001264  011011                                   MOV     (R0),(R1)         ; CLOSE UP LIST
 001266  001002                                   BNE     25$               ; IF NE, NOT END
 001270  010067  000002G                          MOV     R0,$TMLST+2       ; ELSE MARK NEW END IN LISTHEAD

 001274  011510                           25$:    MOV     (R5),(R0)         ; LINK TEMPLATE INTO NEW SPOT
 001276  001002                                   BNE     27$               ; IF NE, NOT END OF LIST
 001300  010067  000002G                          MOV     R0,$TMLST+2       ; MARK NEW END
 001304  010015                           27$:    MOV     R0,(R5)           ; ...
 001306  000241                                   CLC
 001310  000207                           30$:    RETURN

 001312  000261                           101$:   SEC
 001314  000207                                   RETURN

```

```
    236 000270  000167  000036          50$:    CALLR   UPDDLL          ; UPDATE DLL PARAMETERS AND RETURN
    237
    238 000274  000207                  60$:    RETURN
```

```
                                                                  N  5
SETREM     CREATED BY  MACRO  ON 1-JUL-85 AT 16:37        PAGE 3

MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME       REFERENCES

EPRINT           #4-77
ERROR$           #4-66        5-221       11-474      12-553
PRINT            #4-61
RESRG            #4-55        8-379       12-548
SAVRG            #4-55        8-356       12-505
```

```
                        .TITLE  SETSLT - SET SLT PARAMETERS
                        .IDENT  /V05.00/
                        .ENABL  LC
                ;
                ;
                ; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ;
                ; MODULE DESCRIPTION:
                ;
                ; CFE - SET SLT PARAMETERS
                ;
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;




                        .MCALL  CALLR

```

```
    001550  000401                          BR      80$

    001552  000241                  70$:    CLC

    001554  012667  000000G         80$:    MOV     (SP)+,$WILD     ; RESTORE WILDCARDS
    001560  012667  000000G                 MOV     (SP)+,$STAN     ; RESTORE STATION
    001564  012667  000000G                 MOV     (SP)+,$FLAGS    ; RESTORE FLAGS
    001570  000207                          RETURN

            000001                          .END
```

```
                                                                 N  8
SETSVC    CREATED BY  MACRO  ON 29-JUN-85 AT 05:24     PAGE 1
SYMBOL  CROSS REFERENCE                                CREF    04.00
SYMBOL   VALUE              REFERENCES

CO.KNC   = ******  GX        5-113       5-130       5-201
CS.SVC   = ******  GX        5-128
C.STS    = ******  GX        5-128      *5-154      *5-176      *5-181
ERRCIS   = ******  GX        5-196
ERRSVC     000000 R         #4-94        5-213
ERRUSG     000045 R         #4-95        5-216
EX$ERR   = 000002           #4-93        4-94        4-95
FL.DLM   = ******  GX        5-115
OC.MBL   = ******  GX        5-173
OC.MRC   = ******  GX        5-165
OC.MWI   = ******  GX        5-178
OC.NUM   = ******  GX        5-152
OC.RTM   = ******  GX        5-169
OC.STA   = ******  GX        5-183
OC.USG   = ******  GX        5-120       5-144
SCR        000110 R         #4-98
SF$INC   = 000002            5-148       5-150
SF.ENA   = ******  GX        5-190       5-193
SV$DTE   = ******  GX        5-154
SV$PSZ   = ******  GX        5-176
SV$WSZ   = ******  GX        5-181
SW.ENA   = ******  GX        5-191
SW.INC   = ******  GX        5-122       5-146
SW.OUT   = ******  GX        5-122
S$TFLG   = ******  GX       *5-190      *5-193
S$VDTE   = ******  GX        5-157
S$VFLG   = ******  GX       *5-148      *5-150
S$VNAM   = ******  GX        5-134       5-138
S$VPSZ   = ******  GX       *5-175
S$VRCL   = ******  GX       *5-167
S$VTIM   = ******  GX       *5-171
S$VWSZ   = ******  GX       *5-180
WC.CNT   = ******  GX        5-132       5-203
WC.TRI   = ******  GX        5-136
$CFERR   = ******  GX        5-196       5-213       5-216
$CNTNM   = ******  GX        5-134
$COMP    = ******  GX        5-113       5-130       5-201
$DTE     = ******  GX        5-155
$FLAGS   = ******  GX        5-115
$FNSLT   = ******  GX        5-186
$FNSTA   = ******  GX        5-188
$MBL     = ******  GX        5-175
$MRC     = ******  GX        5-167
$MWI     = ******  GX        5-180
$OPRND   = ******  GX        5-120       5-144       5-152       5-165       5-169       5-173       5-178       5-183
$RTM     = ******  GX        5-171
$SESVC     000112 RG        #5-112
$STAN    = ******  GX        5-138
$SWICH   = ******  GX        5-122       5-146       5-191
$TMLST   = ******  GX        5-118       5-185
$WILD    = ******  GX        5-132       5-136       5-203
```

```
SETUNT      CREATED BY  MACRO  ON 12-AUG-85 AT 15:02      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE             REFERENCES

$STADR  = ******  GX      6-290
$SWICH  = ******  GX      5-215
$TMLST  = ******  GX      5-160       6-258
$UNTNM  = ******  GX      5-202
$WILD   = ******  GX      5-180       5-192       5-200
.FNSLT  = ******  GX      6-263
```

```
;****
; LOCAL MACROS
;****

        .MACRO  PRINT   TEXT
        MOV     TEXT,R0         ; POINT R0 AT ADDRESS OF MESSAGE TO PRINT
        CALL    $PRINT          ; PRINT MESSAGE
        .ENDM   PRINT

        .MACRO  ERROR$  TEXT
;
; THE ERROR TEXT MUST HAVE THE ERROR STATUS AS THE FIRST BYTE OF THE ERROR
; MESSAGE STRING.
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ERROR MESSAGE
        .ENDC
        CALL    $CFERR                  ; PRINT ERROR MESSAGE
        .ENDM   ERROR$

        .MACRO  EPRINT  TEXT
;
; PRINT TEXT ON ERROR LUN
;
        .IF DIF <TEXT><R0>
        MOV     TEXT,R0                 ; GET ADDRESS OF ASCIZ STRING TO PRINT
        .ENDC
        CALL    $CFER1                  ; PRINT MESSAGE ON ERROR LUN
        .ENDM   EPRINT
```

```
000676  012767  000007  177446          MOV     #DFALTL,MS3BLK   ; ELSE, THIS MUST BE DEFAULT ACCESS
000704  012767  000311' 177442          MOV     #DEFALT,MS3BLK+2 ;      ...
000712  012767  000006  177436  30$:    MOV     #ONEL,MS3BLK+4   ; ASSUME SINGLE COPIES
000720  012767  000276' 177432          MOV     #ONE,MS3BLK+6    ;      ...
000726  132765  000200  000005          BITB    #OF.SMC,C.FLG(R5) ; DO WE SPAWN MULTIPLE COPIES ?
000734  001420                          BEQ     40$              ; IF EQ, NO .. OKAY
000736  116501  000012                  MOVB    C.CPY(R5),R1     ; GET NUMBER OF COPIES TO SPAWN
000742  012700  000362'                 MOV     #BUFR,R0         ; GET ADDRESS OF SCRATCH BUFFER
000746  005002                          CLR     R2               ; ZERO SUPPRESSION
000750  004767  000000G                 CALL    $CBDMG           ; CONVERT NUMBER TO DECIMAL
000754  112720  000056                  MOVB    #'.,(R0)+        ; INSERT A DECIMAL POINT
000760  162700  000362'                 SUB     #BUFR,R0         ; CALCULATE LENGTH OF STRING
000764  010067  177366                  MOV     R0,MS3BLK+4      ; STORE LENGTH OF STRING IN ARGUMENT BLOCK
000770  012767  000362' 177362          MOV     #BUFR,MS3BLK+6   ; POINT AT CONVERTED STRING
000776  012703  000372'         40$:    MOV     #TBL1,R3         ; POINT AT INPUT STRING TABLE
001002  012704  000402'                 MOV     #TBL2,R4         ; POINT AT ARGUMENT BLOCK TABLE
001006  012301                  50$:    MOV     (R3)+,R1         ; POINT AT INPUT STRING
001010  001624                          BEQ     10$              ; IF EQ, END OF TABLE .. LOOK FOR MORE
001012  012402                          MOV     (R4)+,R2                  ; POINT AT ARGUMENT BLOCK
001014  012700  000000G                 MOV     #$BUFF,R0        ; POINT AT OUPUT BUFFER
001020  004767  000000G                 CALL    $EDMSG           ; FORMAT THE STRING
001024                                  PRINT   #$BUFF           ; PRINT THE MESSAGE
001034  000764                          BR      50$              ; REPEAT LOOP
                                ;
                                ; END OF TEMPLATES FOUND
                                ;
001036  005700                  60$:    TST     R0               ; WERE ANY OBJECTS FOUND ?
001040  001004                          BNE     70$              ; IF NE, YES .. OKAY
001042  032767  000000G 000000G         BIT     #WC.OBJ,$WILD    ; ELSE, WAS THIS FOR KNOWN OBJECTS ?
001050  001401                          BEQ     101$             ; IF EQ, NO .. OBJECT NOT FOUND
001052  000207                  70$:    RETURN                   ; RETURN TO CALLER
                                ;
                                ; ERROR CONDITIONS
                                ;
001054                          101$:   ERROR$  #ERROBJ          ; FAILED TO FIND OBJECT TEMPLATE
001064  000207                          RETURN


        000001                          .END
```

```
                                        .SBTTL  DISPLY - DISPLAY TRANSPORT CIRCUIT INFO
                                ;+
                                ; *** - DISPLY - DISPLAY TRANSPORT CIRCUIT INFO
                                ;
                                ; INPUT:
                                ;       ARGUMENT BLOCKS FROM GETTCI
                                ;
                                ; OUTPUT:
                                ;       CHARACTERISTICS ARE DISPLAYED
                                ;
                                ;-

002504  004767  000402          DISPLY: CALL    GETSLT              ; GET INITIAL SLT/STATION INFORMATION
002510  103427                          BCS     40$                 ; IF CS, NO SLT MACRO
002512  016704  175262                  MOV     CURSLT,R4           ; GET SLT DESCRIPTOR ADDRESS
002516  032765  000000G 000000G         BIT     #LF.X2P,S$LFLG(R5)  ; IS THIS AN X25 LINE ?
002524  001020                          BNE     30$                 ; IF NE YES - NOT A CIRCUIT
002526  032767  000000G 000000G         BIT     #FL.MTP,$FLAGS      ; IS THE LINE MULTI-POINT ?
002534  001002                          BNE     10$                 ; IF NE, YES
                                ;
                                ; POINT TO POINT TRANSPORT CIRCUITS
                                ;
002536                                  CALLR   OUTCIR              ; OUTPUT INFO AND LEAVE
                                ;
                                ; MULTIPOINT TRANSPPORT CIRCUITS
                                ;
002542  052767  000000G 000000G 10$:    BIS     #FL.TRB,$FLAGS      ; SIGNAL TRIBUTARY ADDRESS REQUIRED
002550  004767  000026                  CALL    OUTCIR              ; OUTPUT INFO
002554  004736                  20$:    CALL    @(SP)+              ; BACK TO GETSLT AGAIN
002556  103371                          BCC     10$                 ; IF CC, GOT NEW LINE
002560  042767  000000G 000000G         BIC     #FL.TRB,$FLAGS      ; ELSE, CLEAR TRIBUTARY ADDRESS BIT

002566  000207                  30$:    RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
002570                          40$:    ERROR$  #ERRCIS             ; LINE NOT IN SYSTEM
002600  000207                          RETURN
                                ;
                                ; OUTPUT CIRCUIT INFORMATION SUBROUTINE
                                ;
002602                          OUTCIR:
002602  004767  000730                  CALL    FMTLIN              ; FORMAT THE LINE ID
002606                                  PRINT   #LINMSG             ; PRINT THE LINE ID
002616  112767  000054  175422          MOVB    #<',>,SERMSG        ; INCLUDE SERVICE PARAMETER
002624  012703  001624'                 MOV     #TRNTBL,R3          ; POINT R3 AT ARGUMENT BLOCK TABLE
002630  004767  002404                  CALL    PRINT               ; PRINT IT
002634  016703  175140                  MOV     CURSLT,R3           ; EXAMINE CURRENT SLT
002640  032764  000000G 000000G         BIT     #LF.BRO,S$LFLG(R4)  ; BROADCAST CHANNEL
002646  001425                          BEQ     10$                 ; IF EQ, NO
002650  032767  000000G 000000G         BIT     #NF.END,$CNFIG      ; IS THIS AN END NODE?
002656  001021                          BNE     10$                 ; IF YES, DON'T PRINT MAX ROUTERS, PRIORITY
002660  012700  000000G                 MOV     #$BUFF,R0           ; POINT TO BUFFER
002664  012701  000325'                 MOV     #RPRMSG,R1          ; POINT TO STRING
002670  012702  002030'                 MOV     #STABLK,R2          ; POINT TO ARG BLOCK
002674  016312  000000G                 MOV     S$LRPR(R3),(R2)     ; GET ROUTING PRIORITY
002700  016362  000000G 000002          MOV     S$LBRA(R3),2(R2)    ; GET MAXIMUM ROUTERS
```

```
                                                                                      N 13
SHOCIR - LIST CIRCUIT CHARACTER MACRO V05.03b Thursday 29-Aug-85 11:51  Page 13-1
Symbol table

APR     001471R         ERRNST  001242R         OFFL  = 000003          SF$BLK= 000001          S$VPSZ= ****** GX
APRVAL  001520R         EXEC    001333R         ON      001326R         SF$INC= 000002          S$VRCL= ****** GX
CH.DUP= 000001          EXECL = 000010          ONL   = 000002          SF.ENA= ****** GX       S$VWSZ= ****** GX
CH.D11= 000100          EX$ERR= 000002          OPTMSG  000163R         SF.SER= ****** GX       S$WND   000006
CH.ECH= 000002          FL.DLC= ****** GX       OUTCIR  002602R         SHOPVC  004360R         TRIB    001437R
CH.MDT= 000020          FL.DLM= ****** GX       OUTG    001400R         SHOSVC  003672R         TRIVAL  001457R
CH.PAR= 000100          FL.MTP= ****** GX       OUTGL = 000010          STABLK  002030R         TRNTBL  001624R
CH.PRM= 000200          FL.MUX= ****** GX       PERM    001367R         STAMSG  000266R         TYPBLK  001736R
CH.PRT= 000007          FL.SLV= ****** GX       PERML = 000011          SVCBLK  001726R         TYPMSG  000172R
CH.STA= 000040          FL.TRB= ****** GX       PRINT   005240R         SVCM1   000745R         WC.CNT= ****** GX
CH.SYN= 000004          FL.TRI= ****** GX       PROTBL  001550R         SVCM2   000773R         WC.NAM= ****** GX
CH.TYP= 000010          FL.X25= ****** GX       PVCBLK  001754R         SVCM3   001050R         WC.TRI= ****** GX
CO.KNC= ****** GX       FMTDLM  000406R         PVCBL2  001760R         SVCM4   001121R         WC.UNT= ****** GX
CS.CNT= ****** GX       FMTLIN  003536R         PVCBL3  002002R         SVCM5   001173R         X$3DFP= ** *** GX
CS.DDM= ****** GX       FMTLN1  003542R         PVCBL4  002012R         SVC01   001223R         X$3DW = ****** GX
CS.DTE= ****** GX       FMTLN2  003546R         PVCBL5  002020R         SVCTBL  001674R         X25     001351R
CS.PVC= ****** GX       FMT2R5  000402R         PVCBL6  002022R         SV$DTE= ****** GX       X25L  = 000004
CS.SVC= ****** GX       GETSLT  003112R         PVCLIN  000663R         SV$PSZ= ****** GX       $BUFF = ****** GX
CS.UNI= ****** GX       GETSTA  004322R         PVCMSG  000420R         SV$WSZ= ****** GX       $CBTA = ****** GX
CS.UNT= ****** GX       GETTCI  002724R         PVCMS2  000437R         S$ADSZ  000007          $CFERR= ****** GX
CS.X3P= ****** GX       HEADR   000002R         PVCMS3  000504R         S$CTL   000002          $CIRID= ****** GX
CURSLT  000000R         HEADR1  000050R         PVCMS4  000564R         S$DTE   000020          $CNFIG= ****** GX
C.CSR   000012          HELBLK  001746R         PVCMS5  000625R         S$FLG   000010          $CNTNM= ****** GX
C.CST   000014          HELMSG  000224R         PVCMS6  000707R         S$LBRA= ****** GX       $COMP = ****** GX
C.CW0   000006          HWP     001544R         PVC01   000537R         S$LEN   000032          $C5TA = ****** GX
C.CW1   000010          INCM    001410R         PVCSW   001666R         S$LFLG= ****** GX       $DVNAM= ****** GX
C.DPR   000016          INCML = 000010          PVCTBL  001642R         S$LFT   000012          $EDMSG= ****** GX
C.HTM   000020          LF.BRO= ****** GX       PV$PSZ= ****** GX       S$LHTM= ****** GX       $FLAGS= ****** GX
C.LEN = 000024          LF.ENA= ****** GX       PV$WSZ= ****** GX       S$LNK   000000          $FNPRO= ****** GX
C.LTM   000022          LF.SER= ****** GX       P$VCT = ****** GX       S$LRPR= ****** GX       $FNSLT= ****** GX
C.PRM = ****** GX       LF.X2P= ****** GX       P$VLCN= ****** GX       S$OWNR  000030          $FNSTA= ****** GX
C.STS = ****** GX       LINID   000137R         P$VNAM= ****** GX       S$PKSZ  000004          $GTBUF= ****** GX
C.UNT   000004          LINMSG  000124R         P$VOWN= ****** GX       S$RCL   000013          $PRINT= ****** GX
DDCTBL  001610R         L$CNT = ****** GX       P$VPSZ= ****** GX       S$STA   000003          $PSIPT= ****** GX
DISPLY  002504R         L$NAM = ****** GX       P$VSTA= ****** GX       S$TCST= ****** GX       $SHCIR  002046RG
DSAB    001427R         L$TRI = ****** GX       P$VWSZ= ****** GX       S$TFLG= ****** GX       $STAN = ****** GX
DSABL = 000010          L$UNT = ****** GX       RPRMSG  000325R         S$THTM= ****** GX       $TMLST= ****** GX
D$TADR= ****** GX       NF.END= ****** GX       SCRACH  001532R         S$TIM   000014          $UNTNM= ****** GX
D$TLIN= ****** GX       NONE    001343R         SERMSG  000246R         S$TMR   000016          $WILD = ****** GX
ENAB    001420R         NONEL = 000006          SERV    001355R         S$VFLG= ****** GX       $$$   = 000017
ENABL = 000007          NSTLIN  001302R         SERVL = 000012          S$VNAM= ****** GX       $$$1  = 000065R     002
ERRCIS= ****** GX       OFF     001330R

. ABS.  000032    000   (RW,I,GBL,ABS,OVR)
        005272    001   (RW,I,LCL,REL,CON)
TEXT    000104    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10838 Words  ( 43 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:29.16
```

```
                                                   .SBTTL  $SHEXE - SHOW EXECUTOR CHARACTERISTICS
                                           ;+
                                           ; *** - $SHEXE - SHOW EXECUTOR CHARACTERISTICS
                                           ;
                                           ; INPUT:
                                           ;         NONE.
                                           ;
                                           ; OUTPUT:
                                           ;         THE EXECUTOR CHARACTERISTICS ARE DISPLAYED.
                                           ;
                                           ;-

001500                                     $SHEXE::
001500  012701  000215'                            MOV     #HEADR,R1           ; POINT TO THE FORMAT STRING
001504  012700  000000G                            MOV     #$BUFF,R0           ; POINT TO THE OUTPUT BUFFER
001510  012702  000000G                            MOV     #$GTBUF,R2          ; POINT AT ARGUMENT BLOCK
001514  004767  000000G                            CALL    $EDMSG              ; FORMAT THE HEADER
001520                                             PRINT   #$BUFF              ; PRINT THE HEADER
001530  012705  000000G                            MOV     #$TMLST,R5          ; GET START OF TEMPLATE LIST
001534  010504                                     MOV     R5,R4               ; TWICE
001536  010503                                     MOV     R5,R3               ; ..ONCE MORE FOR GOOD LUCK !
                                           ;
                                           ; SEARCH FOR THE ROUTING TEMPLATE
                                           ;
001540  011505                             10$:    MOV     (R5),R5             ; GET NEXT TEMPLATE
001542  001002                                     BNE     20$                 ; IF NE, OKAY
001544  000167  001136                             JMP     250$                ; ELSE, END OF TEMPLATES

001550  122765  000000G 000000G            20$:    CMPB    #CS.ROU,C.STS(R5)   ; IS THIS THE ROUTING TEMPLATE ?
001556  001370                                     BNE     10$                 ; IF NE, NO .. KEEP LOOKING
001560  010567  177672                             MOV     R5,ROUTMP           ; SAVE TEMPLATE ADDRESS
001564  010405                                     MOV     R4,R5               ; COPY START ADDRESS OF TEMPLATES
                                           ;
                                           ; SEARCH FOR THE NODE TEMPLATE
                                           ;
001566  011505                             30$:    MOV     (R5),R5             ; GET NEXT TEMPLATE
001570  001002                                     BNE     40$                 ; IF NE, OKAY
001572  000167  001064                             JMP     230$                ; ELSE, END OF TEMPLATES

001576  122765  000000G 000000G            40$:    CMPB    #CS.NOD,C.STS(R5)   ; IS THIS A NODE TEMPLATE ?
001604  001370                                     BNE     30$                 ; IF NE, NO - KEEP LOOKING
                                           ;
                                           ; SEARCH FOR THE XPT TEMPLATE
                                           ;
001606  011404                             50$:    MOV     (R4),R4             ; GET NEXT TEMPLATE
001610  001002                                     BNE     60$                 ; IF NE, OKAY
001612  000167  001056                             JMP     240$                ; ELSE, END OF TEMPLATES

001616  122764  000000G 000000G            60$:    CMPB    #CS.LLC,C.STS(R4)   ; IS THIS AN LLC TEMPLATE ?
001624  001370                                     BNE     50$                 ; IF NE, NO - KEEP LOOKING
001626  026427  000000G 114224                     CMP     C.PRM(R4),#^RXPT    ; IS THIS THE XPT LLC ?
001634  001364                                     BNE     50$                 ; IF NE, NO .. KEEP LOOKING
                                           ;
                                           ; SEARCH FOR THE ECL TEMPLATE
                                           ;
001636  011303                             70$:    MOV     (R3),R3             ; GET NEXT TEMPLATE
001640  001002                                     BNE     80$                 ; IF NE, OKAY
```

```
                                                                  N 15
**FILE**ID**SHOLIN







  SSSSSSSS  HH      HH    OOOOOO    LL            IIIIII    NN      NN
  SSSSSSSS  HH      HH    OOOOOO    LL            IIIIII    NN      NN
SS          HH      HH  OO      OO  LL              II      NN      NN
SS          HH      HH  OO      OO  LL              II      NN      NN
SS          HH      HH  OO      OO  LL              II      NNNN    NN
SS          HH      HH  OO      OO  LL              II      NNNN    NN
  SSSSSS    HHHHHHHHHH  OO      OO  LL              II      NN  NN  NN
  SSSSSS    HHHHHHHHHH  OO      OO  LL              II      NN  NN  NN
        SS  HH      HH  OO      OO  LL              II      NN    NNNN
        SS  HH      HH  OO      OO  LL              II      NN    NNNN
        SS  HH      HH  OO      OO  LL              II      NN      NN      ....
        SS  HH      HH  OO      OO  LL              II      NN      NN      ....
SSSSSSSS    HH      HH    OOOOOO    LLLLLLLLLL    IIIIII    NN      NN      ....
SSSSSSSS    HH      HH    OOOOOO    LLLLLLLLLL    IIIIII    NN      NN      ....




LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
B  1
C  1
D  1   SETOBJ
E  1
F  1
G  1
H  1
I  1
J  1
K  1
L  1
M  1
N  1
B  2
C  2
D  2   SETPAR
E  2
F  2
G  2
H  2
I  2
J  2   SETPRO
K  2
L  2
M  2
N  2
B  3
C  3
D  3
E  3
F  3
G  3   SETPVC
H  3
I  3
J  3
K  3
L  3
M  3
N  3
B  4
C  4
D  4
E  4
F  4
G  4   SETREM
H  4
I  4
J  4
K  4
L  4
M  4
N  4
B  5
C  5
D  5
E  5
F  5
G  5
H  5
I  5
```

```
J  5
K  5
L  5
M  5
N  5
B  6   SETROU
C  6
D  6
E  6
F  6
G  6
H  6
I  6
J  6
K  6
L  6   SETSLT
M  6
N  6
B  7
C  7
D  7
E  7
F  7
G  7
H  7
I  7
J  7
K  7
L  7
M  7
N  7
B  8
C  8
D  8
E  8
F  8   SETSVC
G  8
H  8
I  8
J  8
K  8
L  8
M  8
N  8
B  9
C  9   SETUNT
D  9
E  9
F  9
G  9
H  9
I  9
J  9
K  9
L  9
M  9
N  9
B 10
C 10   SETX2P
D 10
```

```
E 10
F 10
G 10
H 10
I 10
J 10
K 10   SETX3P
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11   SHOBJ
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12   SHOCIR
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14   SHOEXE
I 14
J 14
K 14
L 14
```

```
M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15   SHOLIN
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16
```